

Intellectual Property Protection for Software

SEI Curriculum Module SEI-CM-14-2.1

July 1989

Pamela Samuelson
University of Pittsburgh
School of Law

Kevin Deasy
University of Pittsburgh
School of Law



Carnegie Mellon University
Software Engineering Institute

This work was sponsored by the U.S. Department of Defense.
Approved for public release. Distribution unlimited.

The Software Engineering Institute (SEI) is a federally funded research and development center, operated by Carnegie Mellon University under contract with the United States Department of Defense.

The SEI Education Program is developing a wide range of materials to support software engineering education. A **curriculum module** identifies and outlines the content of a specific topic area, and is intended to be used by an instructor in *designing* a course. A **support materials** package includes materials helpful in *teaching* a course. Other materials under development include model curricula, textbooks, educational software, and a variety of reports and proceedings.

SEI educational materials are being made available to educators throughout the academic, industrial, and government communities. The use of these materials in a course does not in any way constitute an endorsement of the course by the SEI, by Carnegie Mellon University, or by the United States government.

SEI curriculum modules may be copied or incorporated into other materials, but not for profit, provided that appropriate credit is given to the SEI and to the original author of the materials.

Comments on SEI educational publications, reports concerning their use, and requests for additional information should be addressed to the Director of Education, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

Comments on this curriculum module may also be directed to the module authors.

Pamela Samuelson
University of Pittsburgh
School of Law
Pittsburgh, PA 15260

Kevin Deasy
University of Pittsburgh
School of Law
Pittsburgh, PA 15260

Copyright © 1989 by Carnegie Mellon University

This technical report was prepared for the

SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position.
It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Karl H. Shingler
SEI Joint Program Office

Intellectual Property Protection for Software

Acknowledgements

The authors wish to thank Karen Lewis of the Software Engineering Institute for her assistance in formatting and preparing this module. Our appreciation also goes to research assistant Catherine Stachowiak Anderson for her assistance in locating resource materials.

The authors would also like to express their appreciation to the Education Program of the SEI for supporting this work, and, in particular, to Dr. Lionel Deimel, for his commitment to seeing this module through to completion.

Contents

Capsule Description	1
Philosophy	1
Objectives	1
Prerequisite Knowledge	2
Module Content	3
Outline	3
Annotated Outline	4
Teaching Considerations	27
Legal Issues and Software	
Engineering Education	27
Suggested Schedules	27
Depth and Nature of Instruction	28
Exercise	28
Bibliographies	33
Notes on the U.S. Legal System	33
Understanding Legal Citations	34
Availability of References	35
Standard Treatises	35
Other Books on	
Intellectual Property Law	35
Books on Computer Law	36
Articles and Reports	38
Cases	42

Intellectual Property Protection for Software

Module Revision History

Version 2.1 (July 1989)	Minor corrections and formatting changes
Version 2.0 (May 1989)	Approved for publication
Version 1.0 (October 1987)	Complete revision and update
	Draft for public review

Intellectual Property Protection for Software

Capsule Description

This module provides an overview of the U.S. intellectual property laws that form the framework within which legal rights in software are created, allocated, and enforced. The primary forms of intellectual property protection that are likely to apply to software are copyright, patent, and trade secret laws, which are discussed with particular emphasis on the controversial issues arising in their application to software. Also included is a brief introduction to government software acquisition regulations, trademark, trade dress, and related unfair competition issues that may affect software engineering decisions, and to the Semiconductor Chip Protection Act.

Philosophy

Many decisions about the development, distribution, maintenance, and enhancement of software are likely to be affected by constraints imposed by intellectual property laws. Intellectual property law provides a “default setting” of rights allocation when software is created. Licensing or other contracting arrangements may satisfy those who wish to vary the rights allocation arrangements that these laws create. In order to foresee the appropriate manner in which to develop and distribute software, it is important that software developers understand the framework of legal rights and responsibilities within which arrangements for the licensing or sale of their software products takes place.

Although it would be comforting to provide unequivocal answers to all important intellectual property questions, the fact is that the intellectual property law is in the process of evolving to provide adequate and appropriate protection for software. There are many questions for which there are as yet no

clear answers. Still, to understand the open questions will be of value to developers.

To provide software engineers a systematic introduction to intellectual property systems affecting software, the authors have developed a framework for understanding the elements of such systems. This framework has been used throughout this module. First, a discussion of the basic elements of most intellectual property systems is provided. Then, the various forms of intellectual property protection potentially available for or affecting software are examined, to the extent possible, with respect to these basic elements. In addition, critical intellectual property issues affecting software development and use are addressed. A separate section examines the relationships among the various forms of intellectual property protection.

This module is the first of three planned curriculum modules addressing legal issues affecting software engineering. A second module, focusing on software development contracts and standard licensing practices is also available [Samuelson88b]. A third module, exploring warranty and tort liability issues arising from the development and maintenance of software, is still in the planning stage.

Objectives

The objective in teaching about intellectual property protection for software should be to provide the software engineering student with sufficient understanding of the forms of intellectual property protection and their relationship to software to enable him or her to understand the constraints such laws may impose on all aspects of the software development process. While it is reasonable to aim for achieving a basic understanding of how some of these intellectual property issues might be resolved, the pri-

mary purpose of presenting the material to future software professionals is to sensitize them so that they will be able to identify situations in which consultation with an intellectual property lawyer is advisable, rather than to prepare them to solve legal problems on their own.

Prerequisite Knowledge

This module is intended as an introductory overview of the intellectual property laws affecting software. In order to understand the concepts presented, it is not necessary that the student have any particular legal knowledge or background. General knowledge of software and software development practices, however, is assumed.

Module Content

Most references in the text are to items in the several bibliographies at the end of the module (see table of contents). For example, [Conley85] refers to a journal article listed under “Articles and Reports.” Citations of court cases are shown in italics (*e.g.*, [MeadData]) and are compiled in a bibliography labeled “Cases.” References to federal law (*e.g.*, [15 U.S.C. §1127]) are made in the text without specific bibliographic entries. Citations of federal regulations are handled similarly (*e.g.*, [37 C.F.R. §1.96]). A discussion of legal citations, along with necessary background information on the U.S. legal system and suggestions of where to find legal research materials, may be found at the beginning of *Bibliographies*.

Outline

1. Intellectual Property and Intellectual Property Systems
 - 1.1. Why Intellectual Property Systems Exist
 - 1.2. Public Domain
 - 1.3. What an “Intellectual Property” Is
2. The Forms of Intellectual Property Protection Affecting Software
 - 2.1. The Copyright Law
 - 2.1.1. Subject Matter
 - 2.1.2. Requisites
 - 2.1.2.1. Subject Matter Requisites
 - 2.1.2.2. Authorship Requisites
 - 2.1.2.3. Procedural Requisites
 - 2.1.3. Exclusive Rights
 - 2.1.4. Limitations on Exclusive Rights
 - 2.1.4.1. Duration of Protection
 - 2.1.4.2. Special Computer Program Provision
 - 2.1.4.3. Fair Use
 - 2.1.4.4. First Sale Rights
 - 2.1.4.5. Private Use Rights
 - 2.1.5. Infringement Standard
 - 2.1.5.1. Standard Copyright Doctrine
 - 2.1.5.2. Scope of Protection
 - 2.1.5.3. Software Case Law in General
 - 2.1.5.4. Particular Issues in Software Case Law
 - 2.1.6. Remedies
 - 2.2. Patent Law
 - 2.2.1. Subject Matter
 - 2.2.2. Requisites
 - 2.2.2.1. Subject Matter Requisites
 - 2.2.2.2. Registration Process
 - 2.2.2.3. Notice
 - 2.2.3. Exclusive Rights
 - 2.2.4. Limitations on Exclusive Rights
 - 2.2.4.1. Duration of Protection
 - 2.2.4.2. First Sale Rule
 - 2.2.5. Infringement Standard
 - 2.2.5.1. Equivalents Standard
 - 2.2.5.2. Software Questions
 - 2.2.6. Remedies
 - 2.3. Trade Secret Law
 - 2.3.1. Subject Matter and Requisites
 - 2.3.2. Exclusive Rights and Infringement Standards
 - 2.3.3. Limitations on Exclusive Rights
 - 2.3.3.1. Duration
 - 2.3.3.2. Reverse Engineering
 - 2.3.3.3. Independent Development
 - 2.3.3.4. Restriction of Employees’ Rights by Employers
 - 2.3.4. Remedies
 - 2.3.4.1. Enforcement against Those Who Misappropriate Trade Secrets
 - 2.3.4.2. Enforcement against Third Parties
 - 2.4. Federal Acquisition Regulations
 - 2.4.1. Subject Matter of Government Procurement Regulations
 - 2.4.2. Treatment of Software, Hardware, and Technical Data under the Federal Acquisition Regulations
 - 2.4.3. Rights Obtained by the Government
 - 2.4.3.1. Unlimited Rights in Government-Funded Software
 - 2.4.3.2. Limited/Restricted Rights in Privately Funded Software

- 2.4.3.3. Rights in Mixed-Funding Software
- 2.4.4. Constraints on the Rights of the Government
 - 2.4.4.1. Practical Constraints
 - 2.4.4.2. Legislative Constraints
- 2.4.5. Government Rights and Traditional Intellectual Property Remedies
- 2.5. Trademarks and Unfair Competition
 - 2.5.1. Trademarks
 - 2.5.1.1. Subject Matter and Requisites
 - 2.5.1.2. Additional Requisites
 - 2.5.1.3. Exclusive Rights
 - 2.5.1.4. Limitations on the Exclusive Rights
 - 2.5.1.5. Infringement Standard
 - 2.5.1.6. Remedies
 - 2.5.2. Trade Dress Protection
 - 2.5.3. Unfair Competition
- 2.6. Semiconductor Chip Protection Act
 - 2.6.1. Subject Matter
 - 2.6.2. Requisites
 - 2.6.2.1. Originality
 - 2.6.2.2. Registration
 - 2.6.3. Exclusive Rights
 - 2.6.4. Limitations on Exclusive Rights
 - 2.6.4.1. Duration of Protection
 - 2.6.4.2. Reverse Engineering
 - 2.6.4.3. First Sale Doctrine
 - 2.6.4.4. Innocent Purchase of Infringing Chips
 - 2.6.5. Infringement Standard
 - 2.6.6. Remedies
- 3. Interplay among Forms of Intellectual Property Law Affecting Software
 - 3.1. Copyright and Patent
 - 3.2. Copyright and Trade Secret
 - 3.3. Copyright and Federal Regulations
 - 3.4. Copyright and Trademark/Unfair Competition
 - 3.5. Copyright and SCPA
 - 3.6. Patent and Trade Secret
 - 3.7. Patent and Federal Regulations
 - 3.8. Patent and SCPA
 - 3.9. Trade Secret and Federal Regulations
 - 3.10. Trade Secret and SCPA

Annotated Outline

1. Intellectual Property and Intellectual Property Systems

1.1. Why Intellectual Property Systems Exist

The primary purpose of the intellectual property laws is to encourage the development and dissemination of innovative works for use by the public. The creation or invention of useful items and artistic works generally requires the investment of considerable time, energy, and resources by skilled, talented people. To encourage such activities, the intellectual property laws provide, as an incentive, the opportunity to obtain exclusive rights to control commercial exploitation of the innovative or artistic work for a specified period of time. The public receives one benefit when the new work is made available to them for use through some form of sale or licensing arrangement. The public obtains a second benefit when, after the expiration of the period of time of exclusive rights protection, the work falls into the public domain and may be freely used and duplicated, thereby providing even greater availability to the public [OTA86, *Mazer*, *Sony*, *Patterson*87, *Davidson*86b].

1.2. Public Domain

Works that are not protected or protectable within an intellectual property system are said to be in the public domain. “Public domain” means that no one holds rights to exclude members of the public from using, altering, replicating, sharing, or commercially exploiting copies or versions of the work that they have legitimately obtained. The work is available for general use by the public without intellectual property restrictions upon that use.

A work may come into the public domain in various ways. For example:

- The work may not qualify as an appropriate subject matter for intellectual property protection.
- Those eligible to assert an intellectual property interest in the work may choose not to make such an assertion or may fail to obtain protection within the appropriate time period or under the approved procedures.
- The duration of intellectual property protection for the work may have expired.

Once a work has entered the public domain, it may be used by any member of the public and remains ineligible for intellectual property protection. If a member of the public finds a use for it, that person cannot obtain intellectual property protection for the work; however, if he or she improves upon it, that improvement alone may be protectable [Lange81, OTA86].

1.3. What an “Intellectual Property” Is

Intellectual property law provides the framework within which legal rights in software are created, allocated, and enforced. As the term itself suggests, intellectual property law provides property rights of a sort in the intellectual products of the human mind. The purpose of these laws is to provide incentives for developing innovations that may contribute to the growth of knowledge and science. There are several types of intellectual property protection that might be available for software, including copyright, patent, and trade secret laws.

Intellectual property systems may be thought of as consisting of six elements:

1. A definition of the subject matter to which the intellectual property law applies (*e.g.*, machines are within the subject matter of patent, but not copyright).
2. A set of requisites for protection, which includes:
 - What qualities the subject matter must possess to be protectable (*e.g.*, how much creativity must be shown to be entitled to intellectual property rights).
 - Who is entitled to assert the intellectual property right.
 - What procedural steps must be taken to acquire or retain the intellectual property rights.
3. A set of rights (“exclusive rights”) to exclude other people from certain activities.
4. A public policy limitation on the extent of the owner’s intellectual property rights.
5. A procedure for determining whether “infringement” has occurred. (An infringement is a violation of one of the exclusive rights.)
6. A specification of what remedies are available.

2. The Forms of Intellectual Property Protection Affecting Software

Although there are some intellectual property systems—such as the Plant Variety Protection Act—that do not apply to software, there are many that do. Many articles, books, and legal decisions discuss or hypothesize about the appropriate forms of intellectual property protection for computer programs. Unfortunately, there is as yet little certainty in this area of the law. Lawyers and legal scholars debate not only the present state of the law, but also the directions in which the law should be moving. Software, which is both a “writing” (traditionally copyright-protected) and a “machine” (traditionally patent-protected), has created particularly difficult accommodation problems for copyright and patent law. (For sources discussing reasons software

has created such difficulties, see [Kidwell85], [Raskind86], [Samuelson84], [Samuelson86a], [Stern86], and [OTA86].)

More attention will be devoted to copyright law in this module than to other forms of intellectual property protection because that is where most of the “action” is at the moment—that is where the most controversial issues are being fought out. In addition, software firms are increasingly relying on copyright as the primary form of intellectual property protection for their software.

Although some early software patent cases were somewhat hostile to the patenting of software, there has been a revival of interest in and use of this form of protection. Some important and open questions about the extent of patent protection for software inventions remain.

After copyright law, perhaps the next most commonly used form of intellectual property protection for software is trade secret law. Although it has some disadvantages for mass-marketed products and although some troublesome questions remain—particularly about the coexistence of trade secret and copyright or patent protection—trade secret protection has many desirable features for software developers. It is especially desirable because it can be used to protect ideas and other valuable information that patent and copyright law may not protect, and it is of potentially unlimited duration.

Because so much valuable software is either developed under federal government contract or sold to the federal government, it is important for many software developers to understand government procurement regulations regarding software. Like traditional intellectual property laws, these regulations allocate rights between developers and users/customers. They may significantly affect software engineering decisions, particularly as to use of proprietary items in software intended for sale to the government. Because it is much more difficult to vary the terms of government contracts with respect to software rights allocations than it is those of contracts not involving the federal government, there is strong incentive for those who deal with the government to understand the principles of the acquisition regulations.

There are other forms of intellectual property protection relevant to software. For example, certain aspects of software can be protected under trademark, trade dress, and unfair competition law. In addition, semiconductor chip designs can be protected under the Semiconductor Chip Protection Act.

2.1. The Copyright Law

Although the Copyright Office began accepting source code listings as copyrightable subject matter as early as 1964, it was not until 1980 that Congress

explicitly added machine-readable computer programs to the subject matter of copyright [CONTU79, Samuelson84]. Copyright is now a commonly relied upon form of intellectual property protection for software.

2.1.1. Subject Matter

Under the United States Constitution, the subject matter of copyright law is the “writings” of “authors.” Article I, section 8, clause 8, of the Constitution empowers Congress to grant exclusive rights to “authors” for their “writings” for limited times, in order to promote the progress of science and the arts. Under §102 of the present Copyright Act, copyright protection is available for “original works of authorship,” such as books, paintings, motion pictures, and sound recordings [Kaplan67]. In spite of the addition of explicit provision for software copyright in the most recent revision of the Copyright Act [17 U.S.C. §101, *et seq.*], some have questioned whether computer programs are a proper subject matter for copyright law [Samuelson84, Davidson83, OTA86].

2.1.2. Requisites

2.1.2.1. Subject Matter Requisites

In order to qualify for copyright protection, a work of authorship must meet certain requisites. Some of these relate to the subject matter itself, which is required to be all of the following [17 U.S.C. §102]:

- “Original”
- “Fixed” in a tangible medium of expression
- A non-utilitarian work

2.1.2.1.1. Originality

“Originality” is not defined in the statute. Congress chose to let the courts define it through common-law refinement. Unfortunately, different court decisions have defined it different ways. In general, originality has been considered a low-level standard, requiring, at a minimum, that the work owe its origins to the person who claims rights in it and that it not have been copied from another. Some cases suggest that a degree of intellectual labor, judgment, or creativity may also need to be shown.

Originality has become a very controversial topic of copyright law, particularly in software matters. Should there be a higher originality standard for certain kinds of works? This question is addressed for software in [Raskind86], [Catalda], and [Gracen]. [MeadData], [Finacial], and [OTA86] treat questions of originality of computer data-

bases. (Not all arrangements of data may be deemed original.) [Samuelson86b] discusses the originality problem with respect to non-human authors in the case of computer-generated works.

2.1.2.1.2. Fixed in a Tangible Medium of Expression

Section 101 of the copyright law defines the word “fixed.” When a work has been embodied in some tangible form (*e.g.*, written on paper), it has been fixed, as long as it is sufficiently stable in its embodiment so that it can be perceived [OTA86]. (A live jazz performance is uncopyrightable because it has not been “fixed.”)

2.1.2.1.3. Non-Utilitarian Work

Useful articles, such as machines, are generally excluded from the copyright realm. Things are considered useful if they do more than convey information or display an appearance. Software seems to be an exception to this general rule [Samuelson84, Karjala87, OTA86]. In challenges to software copyright on the basis of its utility, operating systems have been held to be copyrightable [Franklin], as has microcode [NEC].

2.1.2.2. Authorship Requisites

Generally, it is only the author of a work who may claim a copyright in it. Under the “work made for hire” copyright rule, however, a copyright in work made by an employee within the scope of his or her employment is owned by the employer, even though the employee actually authored the work [17 U.S.C. §201].

There are a few categories of specially commissioned works that are also considered “works made for hire.” In general, software does not qualify under this specially commissioned work exception. Copyright ownership in most specially commissioned software products belongs to the developer of the software, unless there is an agreement in writing reflecting a different allocation of rights.

Ownership of a copyright can be sold, licensed, or simply given to another by the owner. The owner can also license or give up any or all of the exclusive rights of copyright [Samuelson88b].

2.1.2.3. Procedural Requisites

The Copyright Act long required that a copyright notice be placed on a work to be protected. For works created after March 1,

1989, however, no notice is required. Even so, notice may still be advisable, in order to prove willful infringement. Registration of the copyright with the Copyright Office is required for U.S. works in order to sue for infringement.

2.1.2.3.1. Notice

Copyright protection arises for an original work of authorship from the moment it becomes “fixed” in a tangible medium of expression [17 U.S.C. §202]. Copyright protection can be either claimed or waived.

To assert a copyright interest in a work that is or is about to be published (*i.e.*, distributed to the public), it has been traditional to place a copyright notice on the work in an appropriate place. (The Copyright Office has rules about where the copyright notice should be placed.) Failure to affix a notice on works created before March 1, 1989 may cause the work to fall into the public domain, although some inadvertent omissions of notice can be corrected. The copyright notice consists of these elements:

1. The word “copyright” or the symbol “©”
2. The name of the copyright holder
3. The year of publication

For works created after March 1, 1989 notice is no longer required.

2.1.2.3.2. Registration

Registration of the copyrighted work with the Copyright Office is not required in order to have a copyright interest in a work. However, having a registration certificate—which can be obtained only from the Copyright Office—is necessary for U.S. works in order to bring a lawsuit for infringement. It is generally possible to obtain the necessary registration after publication, should the need arise to sue for infringement. Registration, therefore, need not be routinely obtained at the time of publication. There are, however, some limitations on rights and remedies if one does not apply for registration within five years of publication. Obtaining a registration certificate is a simple and inexpensive process, involving only a rudimentary review to determine the work’s “originality.”

Registration requires the deposit of a copy of the work with the Copyright Office. In the case of software, there are now a variety of options for deposit, the most common

being a deposit of the the first and last 25 pages of the source code listing of the program to register it, in lieu of the entire program [Samuelson84].

2.1.3. Exclusive Rights

To own a copyright is to have a set of five exclusive rights with respect to the copyrighted work. (This means the author can prevent others from doing these five things with respect to it.) These rights, set forth in §106 of the Copyright Act, are the following:

1. To reproduce the copyrighted work in copies or phonographic records.
2. To prepare “derivative works” based upon the copyrighted work. (The term “derivative work” is broadly defined in the statute. Establishing a work to be a derivative work has been construed to require showing the taking of protected expression. The right to prepare derivative works has been considered coextensive with the “reproduction” right [Samuelson86a].)
3. To distribute copies or phonographic records of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending.
4. In the case of literary, musical, dramatic, and choreographic works, pantomime, and motion pictures, and other audiovisual works, to perform the copyrighted work publicly.
5. In the case of literary, musical, dramatic, and choreographic works, pantomime, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly.

Software is said to be a “literary work” [Franklin] within the meaning of a very broad statutory definition of the term in §101. Much of the controversy about software protection centers on whether the same rules that apply in cases involving novels and plays should be used for software, or whether software should be treated as a functional work, which would make its scope of protection narrower.

2.1.4. Limitations on Exclusive Rights

The copyright statute sets forth a number of limitations on the copyright holder’s exclusive rights, many of which affect software.

2.1.4.1. Duration of Protection

For authors who are natural persons (*i.e.*, individual human beings), copyright lasts the life of the author plus 50 years.

For corporate authors, copyright lasts for 100 years from creation or 75 years from first publication, whichever is less [17 U.S.C. §302].

2.1.4.2. Special Computer Program Provision

Section 117 of the Copyright Act gives the owner of a copy of a copyrighted computer program the right to:

- Copy the program in order to execute it.
- Make backup copies of it.
- Make some adaptations to it, in order to make it more useful.

This provision is very controversial [Samuelson87, Stern85b, *Vault*].

2.1.4.3. Fair Use

Section 107 of the copyright law permits the “fair use” by others of a copyrighted work. Among the uses that tend to be considered “fair” are criticism, comment, news reporting, teaching (including classroom use of multiple copies), scholarship, and research. Whether material qualifies for “fair use” treatment depends on the purpose for which the material is used (nonprofit educational uses are favored, while commercial use is disfavored), the nature of the copyrighted work, the amount and importance of the section used with respect to the work as a whole, and the effect of the use upon the potential market or value of the work. These factors are considered on a case-by-case basis in determining if there has been an infringement of the copyright [Nimmer86].

Examples of fair use of copyrighted software might include [Raskind86]:

- Use of portions of another’s work to show how it supports the author’s own work (use of a section of code to show how it will work with the author’s own code in performing some desired function).
- Use of sections of another’s work for comment or criticism of it (use of a section of code to illustrate that the author has “a better way” of performing a function).
- Copying sections of another’s work for scholarly discussion (distributing a procedure to a class for discussion).
- Use of sections of another’s work to illustrate its impact (using someone’s code to show that they have achieved a noteworthy breakthrough in the field).

2.1.4.4. First Sale Rights

The “first sale” doctrine of §109 of the Copyright Act permits the owner of a copy of a

copyrighted work to sell or otherwise dispose of that copy (or to publicly display it) without seeking permission of the copyright holder [Straus]. Owning a copy does not, however, give the owner the right to make additional copies of it.

This doctrine does not apply to someone who is merely renting, leasing, or borrowing a copy of the work. (Thus, if someone legitimately purchases a copyrighted software package, he or she can not only make use of that package, but also sell it, give it away, or allow others to watch it run. The person can even authorize someone else to do these things. The purchaser cannot, however, make copies of the software in order to do any of these things unless such copies are permitted under some other exception to the copyright holder’s rights.)

2.1.4.5. Private Use Rights

There may in some instances be what may be termed “private use rights,” that is, rights to use the copyrighted work as one sees fit, within the privacy of one’s own home [Sony, OTA86, Patterson87, Samuelson87].

2.1.5. Infringement Standard

The most controversial and unsettled issues in software copyright law are, unfortunately, the ones that matter the most, namely what standard of infringement should be used in software copyright cases and what aspects of software are within the protection of the copyright. Before considering software copyright issues specifically, we must first consider general copyright standards and methods of analysis.

2.1.5.1. Standard Copyright Doctrine

To win a copyright action, a plaintiff must show that:

1. He or she owns a valid copyright and
2. One of the exclusive rights of the copyright has been interfered with [17 U.S.C. §501, 106].

Ownership of a valid copyright must be demonstrated by obtaining a registration certificate from the Copyright Office. The most usual infringement claim is for unlawful copying. In software cases, it is sometimes the making of a derivative work. (See [Samuelson86a] regarding the difference between copying and derivative work rights.)

Since there is often no admission of guilt from the defendant, case law has developed a two-step procedure for determining whether unlawful copying has occurred. Such a determination requires that there be (1) circumstantial

evidence of some use of the plaintiff's work and (2) circumstantial evidence of an unlawful appropriation of copyrighted expression by virtue of substantial similarities in expression.

To show use of the plaintiff's work, access to it by the defendant must be shown. "Striking similarities" between the works may create a basis for inferring access, even when there is no clear proof of it. Additionally, substantial similarities between the two works must be established, usually through expert testimony and by "dissective analysis" of similarities and differences between the works.

Expert testimony is usually inadmissible as to establishing unlawful appropriation. Instead, courts have conventionally relied on "lay observer" impression as the final determinant of substantial similarity.

See [Arnstein], [Sheldon], and [Nichols]. For a variant analytic process, see [Krofft] and [Roth].

2.1.5.2. Scope of Protection

Generally, copyright protects not the ideas expressed in a work, but only the particular expression used to communicate those ideas. This distinction is not clear-cut, of course, particularly when one considers software. The scope of protection afforded by copyright varies with the subject matter, being greatest where the freedom of expression of the author is least constrained. The protection copyright provides for works of art, for example, is very broad. It is somewhat narrower for works of fact, and very narrow for works of function. Truly functional works are not protectable by copyright at all [OTA86, Samuelson84, Beardsley, Landsberg, Taylor].

2.1.5.3. Software Case Law in General

Both the scope of protection afforded by copyright law and the analytic process by which infringement is determined is uncertain as of now.

2.1.5.3.1. The Breadth or Narrowness of the Copyright

There are widely divergent views on the breadth of copyright law protection. It is too early to tell which view will prevail. We predict that the courts will chart a middle course. A sampling of viewpoints is provided by the following references:

- Supporting narrow protection: [Goldstein86], [Karjala87], [OTA86], [Plains-Cotton].
- Taking an intermediate view of the

proper scope of copyright protection: [Uniden], [Raskind86], [Nimmer86], [StanfordNote86].

- Supporting broad protection: [Davidson86b], [Conley85], [Whelan].

2.1.5.3.2. The Analytic Process for Determining Infringement

Courts and commentators seem generally agreed that the standard analytic procedure for determining copyright infringement may be misleading in software copyright cases. The lay observer step of the *Arnstein* analytic process is inappropriate because the lay observer would not normally *see* the text of the code.

2.1.5.3.3. Proposed Alternate Software Copyright Infringement Standards

A number of ideas have been put forward as means to determine infringement in software cases, some of which are discussed briefly here.

[Conley85] advocates a conduct-oriented standard. By this reasoning, infringement is found if the defendant has made use of the plaintiff's code.

A "black box" test is proposed in [Davidson86b]. Davidson suggests that anything the user of a program can see by using the program—either its external features that can be observed directly or internal characteristics derived by inference from external appearances—is not protected by the copyright. Looking inside the black box and using its code, however, is unlawful. This is quite contrary to the view that similar screen displays carry a presumption of infringement [Whelan, Broderbund, LoyolaNote87].

[Whelan] suggests general "function" and "necessity" tests be applied to potentially infringing software. The general purpose or function of a program is its unprotectable idea; all else about the program is protectable "expression," unless there is only one way to achieve a particular function, in which case it, too, is an "idea."

[HarvardNote82] and [Karjala87] suggest that infringement should be found only where copying is for commercial purposes.

2.1.5.4. Particular Issues in Software Case Law

Software copyright case law raises a myriad of issues, most of which have not yet been settled definitively. Some of the more important is-

issues are listed below, along with references to court decisions and analyses. For some of these issues, we offer predictions of what we think will become the standard interpretation.

2.1.5.4.1. Logic and Structure of a Program

A fundamental question about software copyright is whether its protection extends beyond the details of the particular programming language instructions used to higher-level program abstractions. Is the design—either high-level or detailed design—protected by copyright? [*Whelan*], [*SAS*], and [*StanfordNote86*] suggest that program logic and structure represents copyrightable expression. [*PlainsCotton*] and [*Karjala87*] suggest otherwise. We predict that low-level structures will be found to be protected by copyright.

2.1.5.4.2. Program Functionality

Although some cases have held program functionality to be protected by copyright, most notably *Whelan*, protection of functionality seems to protect ideas more than it does expression. [*Q-Co.*], [*OTA86*], and [*Goldstein86*] suggest this is inappropriate. We believe the courts will not protect program function through copyright.

2.1.5.4.3. Algorithms and Programming Techniques

Does copyright protect the algorithms and programming techniques used in a program? Several cases suggest they may be protected [*Whelan*, *Softklone*]. This view is supported by Conley [*Conley85*] but questioned by Stern [*Stern86*]. We predict that algorithms and programming techniques will not generally be protected by copyright.

2.1.5.4.4. User Interfaces

A number of cases have involved the protectability of various features of the user interface, the most visible part of a program. Many of these cases have focused on copyright protection of screen displays. Two questions have been raised—are screen displays protected by copyright, and is a copyright needed for screen displays separate from that of the program itself? Video games may be copyrighted twice—their screens may be copyrighted as audiovisual works and their programs as literary works [*Williams*]. [*Broderbund*] asserts that a program copyright covers screens; [*Softklone*] asserts otherwise. In [*Kramer*], audiovisual

copyright infringement was deemed provable by showing similarities in the underlying programs; in [*Whelan*], underlying program similarities were deemed provable by similarities in screens.

Input formats are another topic of litigation. [*Synercom*] suggests that input formats are not copyrightable, whereas [*Whelan*] suggests otherwise. Input formats are probably protectable, but we believe the scope of such protection is likely to be narrow.

Commands represent a special kind of input format. There is some question as to their protectability, with at least one case rejecting this notion [*Softklone*]. The argument is being raised more forcefully in two pending cases [*Lotus1*, *Lotus2*]. The issue appears likely to receive fuller consideration in the *Lotus* cases, although we doubt commands will enjoy copyright protection.

2.1.5.4.5. Computer-Generated Works

Works whose “author” is a computer program pose special problems for copyright law. Issues and cases involving computer-generated works are treated in [*Samuelson86b*].

2.1.5.4.6. Protection against Particular Uses

Copyright law may or may not protect the copyright holder against certain uses of a program. For example, is reverse engineering of a program legal under copyright law? It is likely to be, although there is evidence both for [*Davidson86b*] and against [*Hubco*, *SAS*] this position.

Modification of a program by another is use that may be affected by copyright. Probably noncommercial user modifications will not infringe copyright. See [*Samuelson87*], [*Nimmer86*], and [*Stern85a*]. [*Artic*], [*Strohon*], and [*Gilliam*] assert that user modifications are not lawful, however. Third-party rights (*i.e.*, rights of parties other than the developer or user) are treated in [*Samuelson87*] and [*Stern85*].

The copyright law explicitly allows users to make backup copies of software. This provision has led to “shrink wrap” restrictions on copying (restrictions listed on the package and purportedly accepted upon the opening of the package or through some similar act) being declared unenforceable [*Vault*]. However, contributory infringement has been found where a third party sold a device

useful for duplicating video games [*Atari*]. [*Formula*] similarly deals with restrictions on third-party implementation of the right to make backup copies. Stern argues in [*Stern85b*] that this right to make backup copies should be broad.

Translations of programs generally result in unlawful derivative works [*Whelan, Samuelson86a*].

See [*Paula*] and [*Samuelson87*] concerning combining a copyrighted program with other works.

2.1.6. Remedies

Remedies for copyright infringement are quite generous for successful plaintiffs. They include [17 U.S.C. §501-504]:

- Monetary damages for any lost profits of the plaintiff.
- Award of the defendant's profits attributable to the infringement. (The statute requires the plaintiff only to prove gross receipts; the defendant must prove deductible expenses.)
- Statutory damages in an amount the court deems just (where lost profits are hard to prove).
- Injunctions.
- Impoundment and destruction of infringing materials.
- Recovery of attorney's fees (discretionary, but generally available).

2.2. Patent Law

For many years, there was doubt that software could be patented [CONTU79]. Just as software may be too much of a machine to fit comfortably into the copyright system, it may be too much a writing to fit comfortably within the patent system. Though some early decisions by the Supreme Court cast doubt on the patentability of software [*Benson, Flook, Dann*], it is now generally accepted that software patents are appropriate, if properly claimed [*Diehr, Abele, Chisum86*]. Many software patents are now being issued [*Maier87*]. Like the copyright law, patent law is evolving to accommodate software inventions. As with the copyright evolution, significant patent questions remain open.

2.2.1. Subject Matter

Section 101 of the patent statute, found in Title 35 of the U.S. Code, defines the subject matter of patent law as any of the following:

- processes
- machines

- manufactures
- compositions of matter

Improvements in any of these are also patentable. (For a general discussion of patents, see [*Chisum87*].)

A number of questions can be raised with respect to software patents. For example, under what category should software be patented? Can software be patented as a machine? Software patents are generally claimed as "processes."

Should software process patents be restricted to software that transforms matter? Such patents are certainly the most readily justified software patents, as they are most like those for other patentable inventions. Some cases suggest they may be the only valid form of process patent for software. For example, a patent for a rubber-curing process implemented through software has been upheld [*Diehr*], and a system for typesetting alphanumeric information has been held to be patentable subject matter [*Freeman*]. On the other hand, an expert system for assisting physicians with medical diagnoses has been held not to be patentable subject matter because the process does not transform matter [*Meyer*]. Chisum has argued that process patentability should not be so limited [*Chisum86*], and there is some case law support for this point of view [*MerrillLynch*].

A perhaps more difficult question is whether algorithms themselves are patentable subject matter. Algorithms have been held to not be patentable [*Benson*], although the scope of this holding has been qualified [*Abele*]. Prominent computer scientist Allen Newell has raised questions about the advisability of patents for software. Chisum argues in [*Chisum86*] that *Benson* should be overruled (*i.e.*, that algorithms should be patentable). Algorithms, of course, are mathematical in nature—certainly in their essence and possibly in the kind of objects they manipulate—and this causes problems for patent law, as mathematical ideas are not, in general, allowable subject matter for patents. Distinctions have been made by the courts between mathematical and nonmathematical algorithms [*Walter*].

2.2.2. Requisites

2.2.2.1. Subject Matter Requisites

To qualify for patent protection, a software invention must be all of the following [35 U.S.C. §101-103]:

- new
- nonobvious
- useful

2.2.2.1.1. Novelty Requirement

To be patentable, an invention must be “new” or “novel.” What patent law chiefly means by “new” or “novel” is that

- The inventor must apply for a patent within one year of the date of the first public or commercial use of the invention (known as the “statutory bar” type of novelty) [35 U.S.C. §102(b)]; and
- The invention must not have been written about previously by other writers, either in domestic or foreign printed publications, or developed by others who tried to patent it or put it to use in this country prior to the invention by the claimant (known as the “anticipation” type of novelty) [35 U.S.C. §102].

The purpose of the requirement for statutory bar novelty is addressed in [Kenyon] and [Choate87]. Notice that experimental uses of an invention do not require patent application within one year [Cali, Smith].

Prior patents, articles, and use are addressed in [Kalman], [Banner], and [Gillman], respectively.

2.2.2.1.2. Nonobviousness Requirement

What makes a process or machine an “invention,” and not just a modest improvement on the state-of-the-art, is that it is “nonobvious” to persons skilled in the art to which it pertains. This requirement has traditionally been the most difficult to satisfy [Graham, Dann, Choate87, Chisum87].

2.2.2.1.3. Utility Requirement

To be patentable, a process (or other subject matter) must also be “useful.” This is generally not a very difficult requirement to satisfy [Manson, Choate87]. It may be difficult to satisfy this requirement for a mathematical algorithm with no known practical application, however.

2.2.2.2. Registration Process

To obtain a patent, the inventor must make a formal application for a patent, which must include:

1. A specification disclosing the invention, that is, a written description of the invention and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable people skilled in the art to which it pertains, or with which it is most nearly connected, to make and use it [35 U.S.C. §112].

2. A written description of the “best mode” contemplated by the inventor for carrying out the invention [35 U.S.C. §112].
3. A set of “claims” that enumerate precisely each inventive feature for which patent protection is sought. These claims will set the “metes and bounds of protection” [35 U.S.C. §112].
4. A drawing of the invention, if a drawing is necessary to understand the invention [35 U.S.C. §113].
5. An oath by the applicant that he believes himself to be the first and original inventor of the process or other invention for which he solicits the patent [35 U.S.C. §115].

The patent application is submitted to the Patent and Trademark Office (PTO) and is confidential until the patent is issued (until the patent “issues”). At that time, the specification, claims allowed, and best mode are made public.

Failure to make adequate disclosure may cause the PTO examiner not to issue the patent, or if issued, the patent may be struck down as invalid on this ground. The PTO examines patent claims carefully and compares them with the “prior art” to determine if the statutory requisites are met.

Patents are expensive and time-consuming to obtain, but they are quite a strong form of protection, protecting an inventor’s interest in ways copyright law does not. For example, independent development of similar works does not infringe a copyright; but the first inventor takes all rights in the patent system. Patentees can control uses of their works, whereas copyright owners generally cannot. Patent protection does not begin until the patent is issued [Chisum87].

The nature of software raises questions about how patent requisites should be met. For example, what constitutes adequate disclosure of a software invention? Source code is generally not required [37 C.F.R. §1.96], but see [White]. In fact, flowcharts may be acceptable [Ghiron]. Adequate disclosure of the best mode of implementing software inventions is treated in [DisclosureNote86] and [Sherwood].

2.2.2.3. Notice

It is not necessary to affix a patent notice to patented works to receive patent protection. Because the availability of certain remedies depends on whether notice has been given, however, it is advisable to affix a notice of patent

protection. Such notices consist of the word “Patent” (or “Pat.”) and the number of the patent. Once a patent application has been filed, the inventor may put a “patent pending” notice on the goods. There are severe penalties for falsely placing patent notices on products.

2.2.3. Exclusive Rights

To own a patent is to have a set of exclusive rights with respect to the patented invention, that is, a set of rights to exclude others from using the invention in certain ways. Patentees have exclusive rights to make, use, and sell the patented invention. The patent law does *not* give exclusive rights to prepare derivative inventions [*Morse*] or to modify the patented item or process [*Wilbur-Ellis*].

In fact, a patent may be issued for an improvement to an existing patented process or machine to someone other than the owner of the underlying patent.

Because patent ownership is defined in terms of rights to exclude others, it is possible for the owner of a patent not to be able to make, use, or sell his patented invention himself. For example, where the patent is for an improvement on an existing process or machine that is itself patented by another, the patentee who can only make his improvement by making the underlying patented invention cannot produce his invention without the other patentee’s permission.

2.2.4. Limitations on Exclusive Rights

There are fewer limitations on exclusive rights under patent law than under copyright law.

2.2.4.1. Duration of Protection

Patents are granted for a 17-year period. Patent rights begin when the patent is issued, not when the patentee first made application. After expiration of the 17-year period, the patentee has no further rights to exclude; the invention falls into the public domain and can be practiced by anyone [35 U.S.C. §154].

It is unlawful to try to extend the duration of the patent by contract. For example, a license granted by the patentee cannot be conditioned on a willingness of the licensee to continue to pay royalties after expiration of the patent [*Brulotte, Lear*].

2.2.4.2. First Sale Rule

The patentee is entitled to exercise exclusive rights control only over the first sale of the item to the public. In particular, the patentee has no right to forbid reselling of the patented item and no right to forbid its resale for a cer-

tain price. Further, the patentee cannot avoid the implications of this rule by “licensing.” The courts look through the form of the transaction to its substance [*MotionPicture*].

2.2.5. Infringement Standard

Patent rights are limited to that which is specified in the patent claims. A patent, unlike a copyright, describes exactly what the owner of the intellectual property interest claims to be his property.

The patent itself is *prima facie* evidence of the existence of a valid patent. This means that anyone who wants to attack the validity of the patent in court (*e.g.*, for lack of invention or inadequate disclosure) will have to produce evidence to overcome the presumption of validity.

The patentee will not, however, be able to claim a broader scope for the invention than the patent claims reflect. A patent claim is like the description of a piece of land in a deed that sets the bounds of the grant that it describes [*MotionPicture, Keystone*].

2.2.5.1. Equivalent Standard

To infringe a patent, another machine or process need not be identical to the patented machine or process. It need only be “equivalent” to it, by which the courts mean it need only perform substantially the same function in substantially the same way to obtain the same result [*Graver*]. If the two works produce substantially the same results, it does not matter if the second manufacturer improves the machine or process [*Atlas*].

2.2.5.2. Software Questions

As yet, there have been few software patent infringement suits. Most software patent litigation has focused on patentable subject matter issues. Because of early decisions casting doubt on the patentability of software, for many years lawyers steered their clients away from software patents. In the future, it is likely that there will be considerable litigation concerning software patents. Among the interesting future issues are the following:

- If software is patented as a process, can a hardwired version of it infringe the patent [*Davidson83*]?
- If a machine is patented and a programmer implements its functions in software, is the software an infringement [*Pennwalt*]?

2.2.6. Remedies

Patent remedies are less generous than copyright remedies in the following ways:

- The plaintiff cannot recover the defendant's profits.
- There is no statutory presumption that requires the defendant to prove deductions from gross sales.
- Recovery of lost profits are limited to profits for units the plaintiff could have made.
- There is no statutory damages provision.
- The most common monetary recovery is for "reasonable royalty" damages.
- Injunctive relief is less commonly available.
- Recovery of attorney's fees is less common, generally being available only when there has been willful infringement.
- There is no provision for impoundment or destruction of infringing materials.

Nonetheless, patent remedies for willful infringement may be substantial, including the trebling of damages in appropriate cases, such as when the patent holder has been severely damaged [Chisum87].

2.3. Trade Secret Law

Software and its associated documentation are often claimed as trade secrets by their developer. Unlike copyright or patent law, trade secret law is state law. Because of this, it varies somewhat from state to state, particularly as to rights of employees to use knowledge obtained from a former job in new work environments, the enforceable scope of restrictive agreements about noncompeting employment, and available remedies. Trade secret law is largely "common law" (case-by-case application of general rules of law), rather than "statutory law," a fact that adds to its uncertain character. This section discusses those principles of trade secret law that are generally applicable [Milgrim87, Bender86, Epstein84].

In order to protect software trade secrets, firms typically require customers to enter into licensing agreements containing provisions specifically designed to protect them. (See curriculum module *Software Development and Licensing Contracts* for more information about software licenses [Samuelson88b].)

Trade secret law has traditionally been considered a tort (injury law) doctrine [Restatement39], but some recent cases have treated it as a property doctrine [Monsanto]. Because its status as an intellectual property doctrine is of more recent vintage than patent or copyright and because of its common-law character, trade secret law is more difficult to discuss within the standard framework we have been using to present overviews of each intellectual property system. Nevertheless, we have made an effort to do so to the greatest extent possible.

2.3.1. Subject Matter and Requisites

A commonly accepted definition of a trade secret includes both its subject matter and its chief requisites. Under this definition, a trade secret is any formula, pattern, device, or compilation of information (subject matter) used in one's business (requisite), which provides one with an opportunity for a competitive advantage over others (requisite), and which is maintained as a secret (requisite) [Restatement39, Bender86, Milgrim87].

A wide variety of software-related things—a complete software product itself, and possibly ideas, algorithms, techniques, software tools, software components, information, and compilations, among other things—may be protectable as trade secrets. Trade secret protection, however, is restricted in a number of ways.

Not every privately held piece of information possessed by a person or a firm is protectable as a trade secret because not all information is used in business as a source of competitive advantage (*e.g.*, financial data is not protectable as a trade secret). Furthermore, to have a trade secret, a person or institution may need to be "in business" [Wollersheim]. To be a trade secret, something may not need to be already providing one with a competitive advantage, so long as there is a reasonable prospect of its supplying such an advantage in the future.

There is no need to designate something as a protectable trade secret prior to litigation, a difference from both copyright and patent law. There is no formal registration process for trade secrets, and as a result, proof of the existence of the trade secret is required of its owner. One needs to be very careful in litigation not to reveal the trade secret in the course of defending it.

Trade secret protection may overlap with copyright protection. This could be the case, for example, where trade secrets are embodied in a writing [Bender86a]. Because patent law requires disclosure of an invention, something cannot simultaneously be patented and be a trade secret, but an inventor may be able to choose between patent and trade secret protection [Kewanee].

Only reasonable measures to maintain secrecy for trade secrets are required [Christopher]. However, reverse engineering is generally a lawful way to obtain another's trade secrets (see below).

2.3.2. Exclusive Rights and Infringement Standards

It is somewhat awkward to speak of exclusive rights and infringement standards in connection with trade secret misappropriation because, as

mentioned above, trade secret law has historically focused on protecting firms from unscrupulous conduct intended to obtain their secrets, not on defining what rights to exclude come with possession of those secrets. Nevertheless, it is fair to say that a trade secret owner has at least these two rights:

1. A right to enforce a confidential relationship, in the course of which the trade secret owner has revealed a trade secret to another. Confidential relations may be created by explicit agreement [Milgrim87] or by implication from the circumstances [Dravo].
2. A right not to be deprived of the trade secret by trespass, fraud, coercion, bribery, or other improper means [Christopher, Milgrim87].

In addition, by its licensing agreement with a customer, a firm can exercise other forms of control over the customer's use and disclosure of the trade secret, thereby extending its exclusive rights. Similar extensions under patent or copyright law may well be unlawful [Epstein84].

2.3.3. Limitations on Exclusive Rights

2.3.3.1. Duration

Trade secret protection is of potentially unlimited duration. As long as something is maintained over time as a secret and continues to serve as a source of competitive advantage to the firm, the trade secret exists and can be protected. Nevertheless, trade secret protection is sometimes spoken of as "fragile" because it can be lost at any time for failure to protect the secret, or for other reasons discussed below.

2.3.3.2. Reverse Engineering

One who obtains a trade secret through reverse engineering of a lawfully obtained artifact may use it without being liable for trade secret misappropriation. Questions have arisen about the application of this principle to copyrighted software. Some people have argued that reverse engineering is not lawful for copyrighted software, either because it necessitates making a copy of the software that may infringe the copyright or because it may breach a license agreement [Grogan84, Conley85, SAS]. Others have said that reverse engineering is lawful for copyrighted software because it is a fair use of the code and thus not an infringement of copyright, even where there is a license restriction against reverse engineering [Q-Co, Vault, StanfordNote86].

2.3.3.3. Independent Development

Like copyright law, trade secret law considers it lawful for someone to develop the same or a

similar valuable work independently and use it to competitive advantage. That person can even patent it and exclude the first developer, who has held it as a trade secret.

2.3.3.4. Restriction of Employees' Rights by Employers

One of the more significant threats to the protection of trade secrets is posed by employees who leave a firm and take company secrets with them. Employers, therefore, often obtain written agreements restricting employees' rights to work for a competitor or make use of information obtained on the job. Such restrictions must be reasonable in scope and duration, however. Some state laws limit an employer's right to restrict employees in this way [Gilburne82, Samuelson88b]. Employees may also be restricted by implied agreements not to disclose trade secrets revealed to them in the course of employment or that the employer specifically directed them to create for the firm [Wexler, Milgrim87].

2.3.4. Remedies

Because trade secret law is a state common-law doctrine, remedies for misappropriation of a trade secret vary considerably from one jurisdiction to another. While monetary damages and injunctive relief are generally available, other remedies of patent and copyright law may not be.

2.3.4.1. Enforcement against Those Who Misappropriate Trade Secrets

Some courts limit relief for misappropriation of trade secrets to damages and injunctive relief for the period of time it would have taken the defendant to develop the trade secrets independently [Epstein84]. Other courts give damages for past infringements and issue injunctions of indefinite duration [Epstein84]. The law is unclear about the proper measure of damages for revealing a trade secret to the public, not just for use by a competitor [PittNote86]. Attorney's fees are rarely recoverable.

2.3.4.2. Enforcement against Third Parties

If a trade secret is disclosed to a third party by a person obligated not to disclose it (*e.g.*, by a new employer) and the third party begins to use the secret in his own business or to reveal it to others, the owners of the trade secret can stop the further use or disclosure if the third party had notice (circumstances of the disclosure may be construed to give notice) that the information was a trade secret and was disclosed without authorization.

Once a lawsuit is brought, third parties may be put on notice that an improper disclosure of a trade secret has been made. If, prior to this time, the third party made an investment decision in good faith without notice of the potential trade secret claim or has otherwise materially changed his position based on it, he will escape liability to the trade secret's owner [Epstein84, Milgrim87].

2.4. Federal Acquisition Regulations

The United States government acquires a considerable volume of software and technical data under federal acquisition regulations. These regulations resemble an intellectual property law because they embody a standard policy to allocate ownership and use rights between government and contractor. These regulations have the force and effect of law. All transactions by which software and its associated documentation are acquired by the government are governed by the general policy provisions and standard contract clauses set forth in these regulations [Nash83, Samuelson86a].

Although government procurement regulations allocate rights and responsibilities in a way similar to an intellectual property law, they are not structured in the same way as other, more traditional intellectual property laws. For this reason, this section discussing procurement regulations will deviate somewhat from the general framework employed above. We have, however, tried to keep this section as parallel as possible to the earlier discussion. In addition, this section will be somewhat more extensive than others in the module because the acquisition regulations are rather complicated and have, in recent years, been in a state of flux.

The primary source of federal acquisition policy, and the standard contract clauses by which the policy is implemented, is the Federal Acquisition Regulation (FAR) [48 C.F.R. §1.000, *et seq.*]. Agencies have the authority to adopt regulations to supplement the FAR policy to the extent necessary to meet each agency's special needs. Policies and standard contract clauses adopted by individual agencies are supposed to be consistent with the FAR policy, although it is possible to find examples of inconsistencies [Samuelson86c].

Many agencies have used this authority to adopt supplementary regulations. For software development purposes, the Department of Defense supplement (DFARS) [48 C.F.R. §201.1, *et seq.*] is most noteworthy. The DFARS is quite different from the FAR.

2.4.1. Subject Matter of Government Procurement Regulations

The FAR is applicable to all procurements made by the government, except those by the Depart-

ment of Defense (DoD), of intellectual property rights in:

- hardware
- software
- technical data

The FAR is broken down into various sections, referred to as "parts." Part 27, entitled "Patents, Data, and Copyright," deals specifically with the acquisition of rights in software and technical data. Part 52 sets forth the standard contract clauses to be used in implementing this policy. Agency supplements to the FAR are structured similarly [Samuelson86a].

In May 1987, a FAR policy with regard to software and technical data was adopted. Prior to that time, the FAR had no substantive policy for software and technical data acquisitions; however, some individual agencies had established their own policies via agency supplements.

The adoption of an entirely new FAR policy in this area was brought about, in large part, by a recognition that government procurement policies needed to articulate and balance the interests of both the government and private industry with regard to intellectual property rights. The newly adopted FAR policy handles both software and technical data acquisitions within the same regulatory provision, but it provides within that regulation for some differentiation as to the treatment of software and technical data.

In October 1988, DoD adopted its most recent policy on acquisitions of technical data. This update includes a separate section relating specifically to the acquisition of rights in computer software. ("Computer software" is defined as computer programs and computer databases; software documentation is treated as technical data.) The DoD is now in the process of drafting a more comprehensive policy, separate from the current technical data policy, for the acquisition of rights in software.

2.4.2. Treatment of Software, Hardware, and Technical Data under the Federal Acquisition Regulations

Software, in its machine-readable form, has characteristics of both hardware and technical data. It is like hardware in the sense that it interacts with a machine (the computer) and causes the machine to do something, to perform a function. Indeed, software is often a replacement for something that could have been hard-wired into the computer itself [Samuelson86a]. Software is also like technical data in that it is, at some level, a set of written instructions. The source code can be made available in a form readable by human be-

ings. Yet, in machine-executable form, software is much more than written instructions. Like hardware, it can perform tasks.

In software, two capabilities are brought together that are very powerful when combined. Software gives its user:

- The power to perform tasks in a way that has traditionally only been available through hardware.
- The flexibility to change, correct, and even improve upon the product in a way that has heretofore only been available in non-functional items such as written documentation.

Because the complex nature of software has made drafting regulations difficult, those who write the government's acquisition regulations have had trouble formulating an appropriate regulatory policy because they have been unable to see software's differences from both hardware and technical data.

More recently, those responsible for drafting and implementing government acquisition policy through the FAR and agency supplements such as the DFARS have come to recognize the need to draft policies that are responsive to these unique technological and economic aspects of software [Samuelson87]. Additionally, they are coming to realize that, in order to take advantage of the modifiability of software, the government needs to have special rights in both the code and its associated documentation.

2.4.3. Rights Obtained by the Government

The laws authorizing government agencies to adopt procurement regulations direct the government to define the respective rights of government and contractor based on who funded the development effort. In general, rights of the government vary based on whether the product was developed at government expense or at private expense.

If the government funds a development effort, it is standard policy for the government to get "unlimited rights" in software. If privately funded, the government gets site-restricted rights in the software. Until recently, a product had to be funded entirely with private funds to be treated as having been "developed at private expense." In an effort to be more equitable in the allocation of rights in and responsibilities for products such as software, Congress has directed that in "mixed funding" arrangements (some private and some government funding), an intermediate level of government rights be established [Packard86, Samuelson87].

2.4.3.1. Unlimited Rights in Government-Funded Software

2.4.3.1.1. Unlimited Rights

When the government has unlimited rights in software, it has a broad license to:

- Use it (FAR and DFARS).
- Duplicate it (FAR and DFARS).
- Release it (DFARS).
- Disclose it (FAR and DFARS).
- Prepare derivative works of it (FAR).
- Publicly display and perform it (FAR).
- Authorize others to do those things the government has the right to do (FAR and DFARS).

The government generally claims unlimited rights in software developed wholly or primarily at government expense. Unlimited rights are not the same as ownership because the government has no "exclusive rights" in the work. Nor are unlimited rights the same as placing the work in the public domain, in that the contractor may retain ownership and the government acquire only a package of rights in it [Samuelson86a].

Government representatives generally believe that unlimited rights are necessary to achieve competition for procurement and for maintenance and enhancement purposes [Deasy88]. Industry representatives are concerned that if the government provides the software or technical data to a competitor, it may destroy the developer's competitive edge in the marketplace and prevent him from recouping his investment in the software product or the tools of its production [Martin87]. As a result, it appears that some software developers are reluctant to do business with the government.

2.4.3.1.2. Copyright Concerns

The FAR gives each government agency the option of permitting software developers to retain a copyright in software developed for or delivered to the government. A claim of copyright by the contractor generally cuts back the scope of the government's rights from unlimited rights to rights to use, duplicate, and disclose the software *for government purposes only* (i.e., no right is acquired to commercialize the software) [Samuelson86a]. The government therefore relinquishes the right to use the software for commercial purposes or to allow others to do so.

2.4.3.1.2.1. Government Control over Contractor Assertions of Copyright

Under the FAR, the contractor must secure the permission of the government in order to obtain a copyright in publicly funded software. Some agency supplements, such as the DFARS, reverse this and permit contractors to routinely obtain the copyright unless the government affirmatively acts to prevent it by invoking what is called the “special works” clause, which purports to vest the right to a copyright in the government.

2.4.3.1.2.2. Government Claims of Copyright

The government sometimes attempts to claim a copyright in software, even though §105 of the Copyright Act would seem to prohibit the government from directly obtaining a copyright in government-funded work. Use of the “special works” clause of the DFARS is an example of this. This manner of obtaining a copyright is risky for the government, however, since it seems to violate §105, possibly making such a copyright unenforceable [Samuelson86a].

The least risky way for government agencies to attempt to claim such copyrights is by taking the copyright indirectly, as is provided for in the FAR. This is done by having the developer obtain the copyright and then requiring the developer to assign the copyright to the government.

2.4.3.1.3. Flexibility to Obtain Less than Unlimited Rights

The present acquisition regulations provide little flexibility to allow government contract officers to obtain less than the standard set of broad rights in government-funded software. This reflects a concern on the part of government policymakers that, given too much freedom, government contracting personnel may not make good decisions about what rights to get. In addition, some government policymakers express concern that the acquisition of less than unlimited rights would create a severe administrative burden for the government in policing restrictions on the government’s rights to the software or data. What seems to be needed are better guidelines to allow government contracting personnel to exercise their flexibility wisely [Martin87].

2.4.3.1.4. CAD/CAM and Software Tools

It is unclear what rights, if any, the government may claim in CAD/CAM (computer aided design/computer aided manufacturing) or software tools used or adapted for use in performance of a government contract, especially those at least partly developed (even if at private expense) under a government contract. The government generally asserts that it needs access to such tools for maintenance and enhancement purposes, and claims rights in such items if they are used in a government-funded development effort. Contractors generally claim that such tools are critical to their survival as a viable business enterprise and that they must be protected from broad claims of rights by the government [Martin87].

2.4.3.2. Limited/Restricted Rights in Privately Funded Software

The government sometimes finds on the market a privately developed software package that meets its needs. Because use of commercially available software packages can be cost-effective and because it takes less time to purchase and adapt such packages, there is currently an increased emphasis on using privately developed software packages within the government [Packard86]. The government typically gets “restricted rights” in such privately developed software.

2.4.3.2.1. Restricted Rights

Restricted rights for software give the government rights to:

- Use the software with the computer for which it was obtained (FAR and DFARS).
- Use the software with a backup computer if the first computer becomes inoperative (FAR and DFARS).
- Copy the software for backup purposes (FAR and DFARS).
- Modify the software and combine it with other software (FAR and DFARS).
- Disclose it to and reproduce it for use by support service contractors (FAR).
- Use it with a replacement computer (FAR).

The Department of Defense acquires restricted rights in privately funded machine-readable code and “limited rights” (*i.e.*, government-wide rights) in technical data under the DFARS.

2.4.3.2.2. Need for Access to Source Code and Proprietary Data

Often a developer will not include source code or other valuable proprietary documentation as part of its agreement as to what is to be delivered to the government. Government personnel often believe that the government needs access to source code and other documentation to maintain and enhance the software, especially in the event that the developer goes out of business, discontinues support of the product line, or does a poor job supporting the software.

Also of concern is the potential need for access to such information and the ability to correct or adapt the software quickly in case of a national emergency.

Industry representatives are concerned that if they permit the government to have access to valuable source code and other technical data containing proprietary information, this material may end up in the hands of competitors. They are also concerned that if government funds are used to modify a privately developed software package to make it suitable for a particular government use, the government may attempt to claim broader rights, based on the use of public funds in the development of the modified product [Martin87, Deasy88].

2.4.3.2.3. Degree of Flexibility in Negotiating for Rights in Privately Developed Software

The government can always negotiate for greater rights in privately developed software than the standard “restricted rights.” Under the FAR, there is also some flexibility to negotiate for less than the full set of restricted rights. Under the DFARS the government can only accept less than restricted rights in software or limited rights in the documentation if permission is obtained to deviate. In practice, even if not entirely precluded, such negotiations for lesser rights seem rarely to occur [Samuelson86a].

2.4.3.2.4. Databases Containing Technical Data

When privately developed technical data are delivered to the government in a database, the FAR says it should be treated as “technical data,” in which the government has limited (*i.e.*, government-wide) rights. The DFARS does not address this issue, but would seem to treat such data as software.

2.4.3.2.5. Subcontractor Concerns

Much of the software developed for the government is produced by subcontractors. However, it is unclear what rights the government obtains if a prime contractor obtains less rights from the subcontractor than the prime contractor has agreed to deliver to the government. It is possible that the government has a legitimate claim only to the lesser set of rights. This may be especially troublesome when the prime contractor has obtained privately developed software from a subcontractor for use in a government-funded system. The government may only have restricted rights in the privately developed component, even though it expects unlimited rights [Samuelson86a].

2.4.3.3. Rights in Mixed-Funding Software

Until recently, the government treated software developed under mixed-funding arrangements (*i.e.*, using both public and private funds) as if it had been developed at government expense. That is, the government claimed the same broad, unlimited rights in mixed-funding software that it claimed in software developed entirely at government expense, even if the government contribution to the development effort was quite small. Many developers felt this to be an inequitable policy and found no incentive to use their own resources in developing innovative software for the government [Packard-86, Samuelson86a, Samuelson87, Martin87].

Recent revisions in the procurement regulations have attempted to address this inequity. Critical questions, however, remain, some of which are discussed below:

- How much private contribution should be required before treating software as having been developed with mixed funds? Should it be a strict percentage of the total cost or should a decision be based on other factors?

The FAR provides guidance to government contracting officers that they should consider the 50% point (half government, half private funding) as indicating that mixed-funding treatment may be appropriate. This approach allows the contract officers considerable flexibility.

The DFARS, on the other hand, provides no mixed-funding alternative for software at present, but it does provide for a structured approach to negotiations for less than unlimited rights (“government purpose license rights”) in technical data, which includes software documentation. Contracting officers are instructed to consider sev-

eral factors, including “contribution of the respective parties,” before agreeing to take less than the standard unlimited rights in technical data in mixed-funding situations. The government’s rights in software can be cut back to “government purpose” rights, however, if the contractor exercises the right to claim a copyright in the software (see section 2.4.3.1.2, above).

- How should rights be allocated in mixed-funding software?

Regulations have tended toward government purpose rights—which are similar to unlimited rights, with the exception that the government obtains no right to commercialize or to empower others to commercialize the product—as appropriate for mixed-funding software.

- What costs should go into determining if funding is mixed?

The issue has arisen whether use of privately developed tools and expertise, as well as other indirect costs, should be considered. These questions appear still to be open.

- At what point is software “developed”?

Industry generally believes software is developed when detailed design specifications have been prepared, whereas the government generally asserts that testing is necessary before software can be considered developed. The FAR does not define “developed”; the DFARS does and regards software to be “developed” when it “exists and is workable.”

2.4.4. Constraints on the Rights of the Government

Government procurement regulations differ from other systems for allocating intellectual property rights and responsibilities in that they are written by the consumer. (Developers do have an opportunity to comment upon regulations before they are adopted, of course.) As a result, some of the procurement regulations have seemed unbalanced in favor of the government, as contrasted with more traditional areas of intellectual property law, which tend to favor the creators of works, emphasizing the need to provide incentives to creative, skilled people to motivate them to continue producing items that will benefit society.

There are, however, constraints on the government’s ability to simply apportion to itself extensive rights in products such as software and technical data.

2.4.4.1. Practical Constraints

The procurement regulations have been a fertile source of disagreement about the appropriate allocation of rights and responsibilities in software and technical data between government personnel and software developers. The government often claims to need a broad set of rights in software in order to maintain and enhance the software or to achieve competition for procurement. Representatives of the software industry generally express concern that these broad claims of rights by the government inhibit their ability to commercialize the technology and thereby recoup their investment [Packard86, Samuelson87, Martin87, Deasy88].

As a practical matter, of course, developers can refuse to do business with the government if the government claims too broad a set of rights. A developer with a good product probably can market the product commercially. If government claims of rights become too onerous, some truly innovative developers simply conclude that doing business with the government is not worth their while. They may continue to do business with the government but may be unwilling to make their best products available to the government, saving them instead for the commercial market.

Thus, if the government wants to have access to the best technology available, an equitable policy, responsive to the interests of both government and industry, is in the best interest of the government. Such considerations serve to temper the desires of the government, as a consumer, to obtain broad rights in products it acquires [Packard86, Samuelson86, Deasy88, Samuelson87].

2.4.4.2. Legislative Constraints

There are also legislative constraints on the government’s regulatory authority:

- Legislation under which Congress gives to agencies the power to adopt procurement regulations generally requires that the government consider the interests of industry in doing so.
- Section 105 of the Copyright Act specifically prohibits the government from directly claiming a copyright in work prepared by a contractor under government contract. (No similar provision prevents the government from claiming a patent in patentable subject matter.)
- Legislation aimed at strengthening small business requires the government to provide incentives to small business to become involved in the government contract arena.

2.4.5. Government Rights and Traditional Intellectual Property Remedies

Remedies generally available under intellectual property law may be unavailable where the government is a party. Injunctions will not issue against the government for copyright or patent infringement. The government is permitted, by federal statute [28 U.S.C. §1498] to infringe copyrights and patents with only an award of damages as a remedy [Samuelson86a]. This injunction prohibition does not apply to trade secrets included in software or technical data delivered to the government, however. Some government officials have claimed that the government does not recognize trade secrets, but only recognizes contracts not to reveal technical data, thus leaving the contractor with only a monetary damage claim for breach of contract. Case law does not support this interpretation [Megapulse, Samuelson86a].

2.5. Trademarks and Unfair Competition

Every software company and software product has to have a name, symbol, logo, or slogan by which it is known in the trade. These are among the things that may be a firm's trademarks. It is worthwhile for software engineers who may be involved with naming schemes for software systems to have at least passing familiarity with some basic trademark and unfair competition doctrines. In addition, there are a great many trademarks, such as "UNIX" and (formerly) "Ada," which are commonly referred to in the software engineering literature. Software engineers may want to know what care is needed in using these terms.

2.5.1. Trademarks

Trademark law is partly federal and partly state law. There is a federal trademark statute (known as the "Lanham Act" and found in Title 15 of the United States Code beginning at §1051), which, in essence, gives nationwide protection to marks that may have been created through state law. Federal protection arises only when a trademark is registered with the Patent and Trademark Office (PTO). State law may concurrently protect these same marks. (Some states have trademark statutes; others protect trademarks through common law.) Although there are differences among the trademark laws of the various states and some differences between federal and state protection, we will focus here on federal law. Because of its national scope, federal trademark protection can be quite desirable. ([McCarthy84] is the best general reference on this topic.)

2.5.1.1. Subject Matter and Requisites

The Lanham Act describes several kinds of marks for which protection is available [15 U.S.C. §1127]:

- Trademarks
- Service marks
- Collective marks
- Certification marks

Each of these is described below.

2.5.1.1.1. Trademark

The most familiar mark is trademark, which is a mark for "goods." A trademark is any word, name, symbol, or device or any combination of them that has been adopted and used by a manufacturer or merchant to identify his goods and distinguish them from those manufactured or sold by others.

A trademark can be owned by only one person or firm, namely the maker or distributor of the goods. Notice that a trademark serves a *source* identification function, not a *product* identification function. "Cereal" cannot be a trademark, whereas "Kellogg's" can be.

2.5.1.1.2. Service Mark

A service mark is a mark used in the sale or advertising of services to identify the services of one person and distinguish them from services of others.

Like a trademark, a service mark can be owned by only one person or firm, namely the service provider.

Software raises an interesting question with respect to trademarks and service marks—is software a "good" or a "service" [Davidson86a]?

2.5.1.1.3. Collective Mark

A collective mark is a trade or service mark that is used by members of a cooperative, association, or other collective group or organization to indicate membership in the organization.

Collective marks are owned by the organization and may be used (with permission) by members of the organization.

2.5.1.1.4. Certification Mark

A certification mark is a mark used on or in connection with the sale of products or services to certify regional or other origin, material, mode of manufacture, quality, accuracy, or other characteristic.

Certification marks can only be owned by a person or firm that does not make the goods or services being certified. Use of the certification mark must be administered by the owner in a fair way.

“Ada” is an example of a certification mark, in this case, owned by the DoD for use in connection with the marketing of Ada compilers certified by the DoD to meet technical standards it has established.

2.5.1.2. Additional Requisites

2.5.1.2.1. Distinctiveness

Whether a word or symbol will be protectable as a trade or service mark depends heavily on how “distinctive” it is as applied to the goods or services. Because the function of a trademark is to identify the source of the goods, not the type of goods, “distinctiveness” refers to how much a particular word or symbol distinguishes its maker’s goods from similar goods made by others. Computer people often seem to have difficulty selecting truly distinctive marks. Choosing a very distinctive name for one’s company or product avoids litigation and a lot of trouble, however.

Common descriptive or generic terms are not protectable as trademarks because they are not “distinctive” of any one manufacturer. Various computer-related marks have been held to be generic, and therefore unprotectable [*TechnicalPublishing, CES, ComputerStore*].

When words that have had trademark significance become the standard terms for the type of product to which they refer, they can become generic and lose trademark protection. “UNIX,” for example, may be in danger of becoming generic, which is why “®” is so often shown with the name; this is an effort to preserve the mark as a source identity.

Descriptive terms are only protectable if they have “secondary meaning” (*e.g.*, a company becomes well-known in the trade by the name, so that the firm acquires an association with the descriptive name). “International Business Machines” is a good example of a firm that started out with a descriptive name that became distinctive through acquiring a strong secondary meaning. “Systems Software, Inc.” is an example of a descriptive name that might take some effort to make distinctive enough to qualify as a trademark.

Suggestive terms (that is, suggestive of some quality or ingredient of the product) are generally protectable without proof of secondary meaning, though the line between suggestive and descriptive marks is sometimes blurred [McCarthy84].

Distinctive terms (*e.g.*, “Macintosh” for a computer) are protectable immediately and without question.

2.5.1.2.2. Disqualifications

For public policy reasons, there are a number of types of words and symbols which cannot become trade or service marks. These are listed in the statute [15 U.S.C. §1052].

2.5.1.2.3. Registration

Registration of a trademark with the PTO is not necessary to have state law protection for it. To have state law protection, all that is necessary is that the mark be used in trade. Registration is necessary for federal protection, however. When registration with the PTO is sought, the PTO searches its records to see if others have used the mark before issuing a certificate of registration.

2.5.1.2.4. Notice

Like patent law, trademark law does not require that the trademark owner put a notice of intent to claim a mark as a trademark on the product or its packaging. As with patent law, however, the availability of certain remedies may be restricted if no notice appears on the goods.

The “®” symbol is used to designate a registered mark. Either “TM” or “SM” is used to begin to educate the public that a firm is claiming a word, phrase, or other device as a mark or as a way of trying to prevent a well-known mark, like “UNIX,” from becoming generic.

2.5.1.3. Exclusive Rights

Trademark owners have the exclusive right to use the mark in commerce in connection with the sale of the same or similar goods or services [15 U.S.C. §1114].

2.5.1.4. Limitations on the Exclusive Rights

2.5.1.4.1. Duration of Protection

Trademarks generally last as long as their owner continues to use them in commerce. Abandonment of a mark by ceasing to use it in commerce can cause a user to lose all rights to it, however. A mark can also be

judged abandoned as a result of use in a way inconsistent with previous usage.

2.5.1.4.2. Noncompeting Uses

In general, use of someone else's trademark as one's own trademark will not infringe the trademark if the two firms' goods are not in competition with each other [*Polaroid, McGregor*]. Multiple factors are considered in determining whether goods are "non-competing," including an assessment of the likelihood of the first trademark user moving into the competitive arena in which the second user operates. Very famous marks tend to get wider protection.

Use of a trademark term in an article or other non-advertising written materials cannot infringe trademark rights because it is not a competing use in commerce.

2.5.1.4.3. Fair Use

If it is necessary to use trademarked words or symbols to describe adequately one's own product, this use may be fair and noninfringing [McCarthy84].

2.5.1.4.4. Functionality

When a mark becomes an integral part of the product and is no longer a source identifier, courts differ as to whether an infringement has occurred. Some cases have upheld protection [*BostonHockey, Morton-Norwich*], whereas others have not [*Pitt, Job'sDaughters*].

2.5.1.5. Infringement Standard

One who by using the same or similar mark on the same or similar kinds of goods causes consumers to be confused about the source of the goods infringes the mark [McCarthy84].

One who provides another with the means by which to infringe, knowing that the other will use those means to infringe the mark (by passing his or her goods off as the trademark owner's goods, for example) is guilty of contributory infringement [*SnowCrest*].

2.5.1.6. Remedies

Trademark law has quite a generous set of remedy provisions [15 U.S.C. §1114-1118].

2.5.2. Trade Dress Protection

The design of the packaging or aspects of the configuration of a product may itself sometimes be protected against imitative copying, on the theory that this "trade dress" has become "distinctive" to a particular producer and signifies that firm to the public.

"Trade dress" is a common-law doctrine that provides trademark-like protection to packaging and product configurations. It may even extend to user features if they are nonfunctional and have acquired secondary meaning [*Itoh*].

In other respects, trade dress protection is sufficiently similar to trademark that no separate discussion is needed here.

2.5.3. Unfair Competition

There are a variety of common-law doctrines and at least one federal statutory doctrine that limit the use one can make of valuable ideas developed and utilized by an innovative competitor.

One may not, for example, expressly or implicitly represent that your product was made by another. This is known as "passing off" [McCarthy84]. It should be noted, however, that this common-law doctrine cannot be used as a substitute for federal intellectual property law; that is, unfair competition law cannot be used to gain protection for an item that does not qualify for patent or copyright protection. The unprotected item is in the public domain and, as such, may be used by another. The common-law doctrines protect instead against deception as to the source [*Sears*].

Similarly, misrepresentations about products or their sources can give rise to an unfair competition action. Federal statutory protection against such misrepresentation is provided under §43(a) of the Lanham Act [15 U.S.C. §1125(a)] [McCarthy84, *Whelan*]. Actions relating to deceptive advertising arise under state doctrine.

Another form of unfair competition is misappropriation—taking a product or item of another and using it for one's own benefit. This doctrine had its origin in *INS*, a case in which the court found misappropriation where one news service took verbatim the uncopyrighted news stories of another [*INS*]. Arguments based on the doctrine of misappropriation of software applications have been raised in several cases but were found to have been preempted by federal intellectual property law [*Videotronics, Synercom*].

2.6. Semiconductor Chip Protection Act

There are at least two reasons why it may be useful for software engineers to know something about the Semiconductor Chip Protection Act (SCPA). First, there may be occasions when they will be working with those who are designing specialized semiconductors for a software system and may find it useful to have some knowledge of the chip law, which differs quite significantly from the copyright law protecting software. Second, the SCPA is a closely related form of intellectual property law that may

one day serve as a model for a new intellectual property system for software. (Programs embedded in chips are not protected under SCPA, but under copyright law.)

2.6.1. Subject Matter

In 1984, Congress passed a new form of intellectual property protection specifically designed to protect rights in semiconductor chip designs, as reflected in the mask works through which chips are created. This law provides protection for developers of a mask work fixed in a semiconductor chip product [17 U.S.C. §902]. Congress, after studying semiconductor chips, determined that chips are sufficiently different from traditionally copyrightable materials such as books, paintings, and motion pictures so as to warrant a separate and unique form of intellectual property protection. SCPA incorporates some features of copyright law and some of patent law, as well as some features specially tailored to deal with semiconductors. It is worth noting that had Congress applied similar reasoning to software, a separate form of protection, other than copyright, would have been warranted in that area also [Kastenmeier85, Kidwell85, Samuelson85, Stern85a].

2.6.2. Requisites

2.6.2.1. Originality

Protection under SCPA is not available if the mask work is not original or if it consists of designs that are “staple, commonplace, or familiar in the semiconductor industry.” This is thought to be a higher originality standard than that of copyright [Stern85a].

2.6.2.2. Registration

To obtain protection under SCPA, the mask work must be registered with the Copyright Office within two years of the first commercial use of the chip—a patent-like feature. If not registered within this time, the mask work will be considered to be in the public domain. Like copyright, registration of a chip involves only a rudimentary review of the originality of the design [Samuelson85, Stern85a].

2.6.3. Exclusive Rights

The owner of a registered mask work has the exclusive rights to reproduce, import, and distribute the mask works and chips embodying the mask work design [17 U.S.C. §905, Stern85a, Samuelson85].

2.6.4. Limitations on Exclusive Rights

The owner of a registered mask work has the right to sue another for infringement if the other

reproduces, imports, or distributes an identical or substantially similar copy of the chip. However SCPA limits the exclusive rights of the registered mask work owner in certain ways.

2.6.4.1. Duration of Protection

The term of protection under SCPA is 10 years. It begins when the registration certificate is issued [Samuelson85].

2.6.4.2. Reverse Engineering

The act permits reverse engineering of the mask work. It is permissible for a person to reproduce the mask work for purposes of teaching about or studying it. Further, one can incorporate what he learns from studying the mask work in another chip without infringing SCPA rights, so long as the resulting chip is not substantially identical to the first chip [17 U.S.C. §906, Raskind85].

2.6.4.3. First Sale Doctrine

SCPA also includes a first sale provision similar to that found in the copyright and patent laws. Under the first sale doctrine, the owner of a lawfully acquired protected chip may use, import, or redistribute that particular chip without concern about potential liability to the owner of the protected design. He or she may not, however, copy the chip for other than permissible purposes [Stern85a].

2.6.4.4. Innocent Purchase of Infringing Chips

One who unknowingly purchases an infringing chip incurs no liability for importing or distributing the chip prior to learning of the infringement. After learning of the infringement, the innocent purchaser will be responsible only for the payment of a reasonable royalty to the design owner [Stern85a].

2.6.5. Infringement Standard

In general, one who produces chips substantially similar to a protected chip may be liable for infringing the mask right. Independent development of the design, however, is a defense. Substantial identity between the protected chip and the allegedly infringing chip has to be shown by the plaintiff. If the defendant has reverse engineered the protected chip, the existence of a credible “paper trail” to support a reverse-engineering privilege is quite important to his case [Raskind85, Stern85a].

2.6.6. Remedies

An award may reflect actual monetary damages suffered, and may also include an award of any

profit realized by the infringing party. SCPA also provides that, at any time before final judgment, the aggrieved party may elect to accept an amount up to \$250,000 as the damage award instead of actual damages or profits. The specific amount in a given case is left to the discretion of the court [17 U.S.C. §911(c)].

A court may also issue an injunction temporarily or permanently prohibiting further infringement and, as with the copyright law, may order the destruction of infringing copies.

In addition, the court has the discretion to award the payment of reasonable attorney's fees to the prevailing party in the suit [Stern85a].

3. Interplay among Forms of Intellectual Property Law Affecting Software

Until quite recently, the subject matter domains of the various intellectual property laws were perceived to be sufficiently distinct that it was relatively rare for firms to claim overlapping protection for their products. Nowadays, claims of overlapping protection are common. Nowhere is this better illustrated than in the case of software [Samuelson85]. When two or more kinds of intellectual property protection are claimed for a product, potential for conflicts among doctrines of the various laws arises and may need to be carefully attended to. The primary areas of potential conflict and related issues are discussed below.

3.1. Copyright and Patent

Until computer software, there was no subject matter of copyright that was simultaneously patentable [Samuelson84]—except ornamental designs for articles of manufacture, which could be covered by either copyright or the design patent statutes [Mazer, Yardley]. Games might be patentable, and graphics of the board layout might be copyrightable as pictures. Engineering designs might be patentable, and drawings of engineering designs might be copyrightable as drawings. Recipes might be patentable, and a compilation of recipes might be copyrightable as a compilation. But the domains of copyright and patent were separable. Here are a few questions that the overlap of subject matters raises:

- Does one have to opt for either a copyright or a patent for software [Kline86]?
- If one can get both a copyright and a patent for software, what does each cover [Maier87, Davidson83]?
- If there are some things that both can cover, what happens when the patent expires? What, if anything, falls into the public domain?
- If there are some things, such as algorithms, that patent law may consider to be unprotectable “ideas,” can copyright protect them [Whelan, Chisum86, Newell86]?

- If a potentially patentable piece of software or aspect of software is not inventive enough to qualify for patent protection, can it nonetheless be protected by copyright?
- If software modules become standard reusable components, can they be patented or copyrighted?

3.2. Copyright and Trade Secret

The legality of applying reverse engineering to copyrighted software to obtain the trade secrets the software contains is a live controversy in the copyright/trade secret overlap. Other questions involving copyright and trade secret interaction include [Bender86, Samuelson84, Davidson86b]:

- If a work is “published” within the meaning of the copyright law, any “ideas” in it—including things that might otherwise be claimed as trade secrets—are, under traditional copyright law, in the public domain. Is this true for software trade secrets? What does it mean for software to be “published”?
- If a copyright notice is affixed to a work whose authors also claim it as a trade secret, what effect does the notice have upon the trade secret?
- If thousands of copies of a piece of copyrighted software have been distributed, can it be said that trade secrets still exist in it?

3.3. Copyright and Federal Regulations

A few examples of the many issues raised by the interplay of copyright and the FAR include:

- Is the “special works” clause by which the government sometimes attempts to acquire the copyright (or some other ownership interest in software) in conflict with the provision (§105) of the copyright law that forbids direct ownership of copyrights by the government?
- May trade secrets exist in published, copyrighted software?
- How does a copyright held by the contractor affect the derivative work rights of the government?

Questions involving copyright and the DFARS include [Samuelson86a]:

- What is the effect on the rights of the government if a contractor copyrights software? Does it reduce the government's rights to a government purpose license?
- Is the DoD special works clause in conflict with §105 of the copyright law?
- What rights, if any, does DoD have to make or authorize the making of derivative works from uncopyrighted software?

3.4. Copyright and Trademark/Unfair Competition

There is some interaction between copyright and trademark/unfair competition law. Some consequences of this interplay are worthy of mention here.

Pictorial trademarks potentially may be protected by both trademarks and copyrights. Furthermore, if a mark is both copyrighted and trademarked, a broader set of exclusive rights over noncompeting goods may be exercised.

Even expired copyrighted material can be recaptured from the public domain if it is used as a trademark.

Unfair competition claims, which are, in essence, equivalent to copyright, may be preempted by copyright law [*Sears, Synercom, Videotronics*].

3.5. Copyright and SCPA

The subject matters of copyright and SCPA do not significantly overlap.

Copyright protection is available for drawings of semiconductor chip designs, but this does not protect the mask work or chips that might implement them. SCPA protection is necessary to protect the chip design.

Copyright protection is available for computer programs encoded permanently or temporarily in chips; SCPA protection of the chip is independent of this.

3.6. Patent and Trade Secret

Because of patent law's disclosure requirements, an invention cannot be both patented and held as a trade secret.

A firm that withholds, in a patent application, material necessary for the specification of its invention can not only have the patent stricken, but may also be liable for fraud on the Patent Office and lose the trade secret [*Colt*].

When the patent on an invention has expired or been invalidated, the invention may not be reclaimed as a trade secret.

Material that need not be included in the patent specification, but which is used in manufacture of the item, may be eligible for trade secret protection.

3.7. Patent and Federal Regulations

Government acquisition regulations extensively regulate the relationship between the government and contractors with respect to patent rights. In the software area, however, the patent/government regulations interface has largely been ignored, primarily because patent lawyers have doubted that software patents would be upheld. As more software patents are issued, the government will need to rethink its software patent policy.

3.8. Patent and SCPA

Although there seems to be some overlap in the subject matter of patent law and SCPA, opportunities for conflict between these laws appear minimal, for it is unlikely that a patent would be issued to protect the whole of a semiconductor circuit design, which is what SCPA protects.

If an inventive portion of a chip circuitry design is patented, can it also be covered by SCPA? If so, what happens when the SCPA protection period expires? If not, can a semiconductor designer seek protection under SCPA after rejection of a patent application?

3.9. Trade Secret and Federal Regulations

There are a number of questions regarding the interactions of trade secret law and federal acquisition regulations [*Samuelson86a*]. For example, can something in which the government has unlimited rights (or government purpose rights) be held by the developer as a trade secret?

By treating all copyrighted software delivered without notice that it is unpublished as "published copyrighted software," can the government claim that all trade secrets in it are dissipated?

Can the government disclose a trade secret in which it has limited or restricted rights without fear of an injunction?

3.10. Trade Secret and SCPA

Because of mask work registration requirements and because of the right to reverse engineer chips, concurrent trade secret and SCPA protection is unlikely. Processes by which SCPA-protected chips are made can be trade secrets, however. Furthermore, until commercial distribution, chip masks and designs can be trade secrets.

Teaching Considerations

Legal Issues and Software Engineering Education

Intellectual property law provides the framework, the default setting in a sense, for allocating rights in software among developers and users of a product. Because the allocation of intellectual property rights determines the legitimate uses that can be made of a software product by both developer and user, an understanding of this area of law is of critical importance to the software engineer.

This curriculum module is one of three originally planned by the Software Engineering Institute covering legal issues related to software. The allocation of rights resulting from the intellectual property laws may be altered by licensing or other laws. The module *Software Development and Licensing Contracts* [Samuelson88b] discusses the types of licensing and contractual arrangements often used to structure the allocation of rights in software products. A third curriculum module on software legal issues discussing principles and concerns relating to warranties and product liability law is not yet under development.

The authors believe that these three legal areas—intellectual property, licensing and contracts, warranties and liability—have significant implications for software development, distribution, and maintenance. Practicing software engineers should, therefore, have at least passing familiarity with these areas. This presents a challenge to software engineering educators that we believe must and can be met.

Law, like other disciplines, has its own terms of art, concepts, and doctrinal rules that may not be obvious to those from other areas of expertise. For this reason, the non-lawyer teaching legal issues may find it expedient to consult with a lawyer regarding materials to be presented. Most communities have an active intellectual property or patent law group affiliated with the local bar association. Considering the timeliness and importance of software legal issues, the instructor should have no difficulty finding an attorney with intellectual property expertise and interest in contributing to instruction in this area. In fact, a school contemplating the teaching of the material presented in this module may wish to consider having an intellectual property lawyer teach it (or participate in team teaching it) on an adjunct basis.

Many communities have a law school with a library in which most of the materials cited here can be found. For those areas where no law school is available, the county law library should provide access to needed resources. At the beginning of *Bibliographies* is information to help the non-lawyer instructor understand and make use of legal research materials.

Suggested Schedules

Obviously, not every software engineering program is ready to devote entire courses to legal issues. This module should be helpful to most instructors, however, despite differences in the amount of instructional time to be devoted to legal issues. Topics can be chosen according to perceived student needs. Some areas of intellectual property law deserve greater attention than do others, and some can, if necessary, be excluded altogether. The following guidelines should enable the instructor to allocate available time appropriately.

Coverage of Legal Issues in One Week or Less.

It would be useful to begin legal issues coverage with topics on basic software development contracts from the beginning of *Software Development and Licensing Contracts* [Samuelson88b]. With a rudimentary understanding of software contracting, the software engineering student will be in a position to better understand the significance of intellectual property principles affecting those contracts. For example, some provisions of the copyright law, such as the “work made for hire” doctrine, provide requisites that must be adhered to in the contracting arena. An understanding of the basics of software development contracts will enhance the student’s appreciation of such copyright doctrines.

The instructor should next move to the area of intellectual property law. In this area, particular emphasis should be given to the material concerning the component elements of intellectual property systems, as well as to copyright law, since it is the system of intellectual property protection which has the greatest impact on the software industry. For those devoting only a week or less to legal issues, these are the core materials that should be covered.

Coverage of Legal Issues in Two or More Weeks.

For those covering legal issues in greater depth, perhaps over a span of a couple of weeks, attention should also be given to the area of trade secret law, since it is the form of protection many developers choose for their most valuable technology (*i.e.*, maintaining that technology and related information in confidence). In fact, it may be possible to protect software by both copyright and trade secret law simultaneously.

Patent law has come to be of considerable importance in the protection of software innovations and will become more important over time. Accordingly, the authors recommend adding it after adding trade secrets. As time permits, the material on government procurement regulations should also be examined, since much innovative software development work is conducted under government contracts.

Among the remaining areas covered in this module, the Semiconductor Chip Protection Act deserves some attention as a form of legislation designed specifically to apply to an innovative technology. Trademark law is probably the area of intellectual property law deserving the least attention, although it may be important with respect to some user interface issues. Following coverage of the intellectual property law area, the instructor could, as time permits, return to the software contracts module for topics, covering the various ways in which the allocation of rights in software may be altered by licenses and other agreements among parties.

Semester-Long Coverage of Legal Issues. For those intending to spend a longer period of time on legal issues, discussion of warranty and product liability issues is recommended. Issues such as whether software should be treated as a good or a service under commercial law can be discussed. This issue is important because, if software is treated as a good, certain implied warranties attach to software products that would not be available if software were treated as a service.

Other Alternatives. Legal issues may, of course, be incorporated successfully into other software engineering courses. For example, the legal issues might be incorporated into a course focusing on software safety, information protection, or software design. Since intellectual property rights so deeply affect decisions at all stages of the software life cycle, this module might be used in conjunction with courses dealing with any stage of the software life cycle, from requirements definition on through maintenance and enhancement.

Depth and Nature of Instruction

In teaching the material included in this module, the instructor may choose to confine him- or herself to the more basic material presented herein, such as the component elements of the various forms of intellectual property protection, and some of the primary intellectual property issues regarding software.

In presenting basic information regarding intellectual property law as it affects software, the instructor should find the exercise included below extremely helpful. Case studies, whether real or hypothetical, are a major tool of legal education. They can be both instructive and engaging and should therefore be considered as a pedagogical tool for teaching legal issues related to software.

The authors recommend reading the cases listed in the bibliographies. Having students make presentations on them in class can be an effective learning experience for students and teacher alike. Doing so can lead to lively classroom discussion of the issues raised.

Students need to realize that intellectual property law is in the process of evolving to provide adequate and appropriate protection for software; there are many important questions for which clear answers do not yet exist. Still, it is of value to understand what the open questions are. Where there is a difference of opinion or some uncertainty as to the law, the authors have attempted to give an indication of the direction in which the law seems to be moving. As time permits, the teacher may want to expand instruction to include some of the more difficult “gray” issues. As the complexity of the issues discussed increases, the instructor may want to consider having an intellectual property lawyer co-teach the course.

Exercise

Emily is a graduate student in software engineering at Module University, a large, private school in the Midwest. Emily has recently written a highly original and useful computer program capable of performing a series of accounting functions commonly used in small professional operations, such as a physician’s office. Emily wrote the program to fulfill the requirements of a class project last semester. It was written using hardware and various software tools made available to students by the university. The program is stored on a disk that Emily pur-

chased herself at the Module University Computer Store.

Module University has an extremely liberal intellectual property policy, which provides that: “In works created by a student in fulfillment of degree requirements, the author or inventor is entitled to claim any intellectual property rights.” Students are permitted to use university facilities for any class projects or assignments.

A. Copyrightability.

1. Is Emily permitted to claim a copyright in the software?

Yes—computer programs are a subject matter qualifying for protection under the copyright act, and from the information given, this program seems to meet the requirements for copyrightability. (It is original and fixed in a tangible medium.) Emily is the author and thus qualifies as the person to claim such protection.

2. If so, how would she go about claiming a copyright?

Copyright subsists automatically in the work.

3. Does she have to put a copyright notice on it?

For works created after March 1, 1989, notice of a claim of copyright is no longer necessary, though it still may be advisable. This is how a copyright notice might look:

© Copyright, Emily (1989)

4. Must it be registered with the copyright office?

The copyright need not be registered with the copyright office to be valid. It would, however need to be registered before Emily could bring a suit against another for infringement of copyright. This could be accomplished at the time Emily intended to bring such a suit.

5. Bernard, another graduate student at Module University, would like to reuse a module or segment of the code Emily has written for a project he is working on this semester.

Assuming Emily has claimed a copyright in the program, could Bernard reuse the module?

The answer to this question is not entirely clear. Generally, reusing a module from Emily’s program in another program involves a potential infringement of two exclusive rights held by Emily as the copyright holder—the exclusive rights to reproduce in copies and make derivative works.

6. What would we need to know to determine whether or not Bernard could reuse some of the code without permission?

If Bernard were going to use only a small portion of the code in which Emily has claimed a copyright, such use may be argued to be a “fair use,” as permitted by §117 of the copyright law.

We also need to determine whether or not the module Bernard wants to reuse was in fact Emily’s original work. If Emily reused a module that was in the public domain or in which another holds a copyright, that module would not fall within the scope of her copyright.

7. Would it make a difference how much, or what proportion, of the code he was reusing?

If the module Bernard wants to reuse is only a small part of the program, there is a greater chance that reuse of the module would be found to be a permissible “fair use.”

8. Would it matter how important the code was to the original program?

The less significant the module is to the copyrighted program, the greater is the likelihood that reuse of that module would be a permissible “fair use.”

9. Would the fact that Bernard wishes to use the software for educational purposes make a difference?

The fact that Bernard wishes to reuse the module for educational rather than commercial purposes increases the likelihood that such reuse would fall within the “fair use” exception of the copyright law.

10. Could Bernard reuse Emily’s detailed design? Her high-level design?

Although the Whelan case suggests that the detailed design would be within the scope of Emily’s copyright, the Plains Cotton decision suggests otherwise. It is also an open question whether the higher-level design is within the scope of copyright protection.

11. Could he rehost, retarget, or translate her program?

Again, this is not an entirely clear issue. Rehosting, retargeting, and translating would, however, most likely be counter to Emily’s exclusive right to make derivative works, and could, therefore, be an infringement of her copyright.

12. Could he base his interfaces on hers?

Again, this is a difficult and not entirely clear question, but probably implicates the derivative work right of the copyright law.

13. Would he need to go to Emily for permission to do any of these things?

Since Emily is the copyright holder, Bernard can copy, reuse, rehost, retarget, translate, or otherwise make substantial modifications to the program with her permission. Bernard might, however, be able to make some modifications to his own copy of Emily’s program so long as he does not distribute the modified program commercially.

B. Patentability.

1. Can Emily obtain a patent on the software?

In some circumstances, a patent can be obtained for an

invention implemented by a computer program. Emily would, under Module University's intellectual property policy, be the one eligible for patent protection if such protection could be obtained.

2. What would she need to show to obtain a patent?

Most likely, Emily would have to show that the program was in fact an inventive process. Current patent office policy does not seem to require that a program process transform matter to be eligible subject matter, but some court decisions suggest that transformation might be required.

3. Would the same showing of originality which would entitle her to a copyright suffice for purposes of obtaining a patent?

No. The patent law requires a higher showing of originality than does the copyright law. To gain a patent, the program must be shown to be "novel," "nonobvious," and "useful."

4. Would the period of intellectual property protection be the same under a patent as under a copyright?

Patent protection lasts for a significantly shorter period of time (17 years) than does copyright protection (generally around 75 years). Therefore, the holder of a copyright will hold a monopoly over the protected item far longer than would the holder of a patent.

5. If she does obtain a patent, could she also copyright the software?

Yes, the patent will be for the function or algorithm (the "idea") in the program; the copyright protects the "expression" of that idea.

C. Company Situation.

1. What if Emily were working as a summer employee on a project for a company and wrote the program as part of her job there.

Could she still copyright it?

Under the "work made for hire" doctrine of the copyright law, the company would be entitled to claim the copyright. The company could, nonetheless, permit Emily, as the author, to claim the copyright.

2. Could anyone copyright it?

Since, under the "work made for hire" doctrine, the company is deemed to be the author of the program for the purposes of the copyright law, the company could claim a copyright in the program if all other requirements, discussed above, are met.

3. Would the situation be different if she were working as a consultant rather than employee of the company? If so, in what way?

The situation would be different because the "work made for hire" doctrine works differently for consultants and special contractors than it does for employees. As a consultant, the author—Emily—would be the one entitled to claim a

copyright in the program. Emily could, however, agree to assign her claim to the copyright to the company.

4. If Emily were able to copyright the program, could the company modify the program as its accounting needs change with the growth of the company?

The company would encounter the same problems as Bernard if it wanted to modify the program after its being copyrighted by Emily.

5. Could Emily patent the software? Why or why not?

Emily would be entitled to attempt to claim a patent in the program under the same circumstances under which she could claim a copyright—for example, if she developed the program while working as a consultant for the company. Of course, the program would have to be otherwise eligible for patent protection for a patent to be issued.

D. Trade Secret.

1. How would Emily go about maintaining a trade secret in the software?

Emily would need to enter a nondisclosure agreement with anyone to whom she licensed the software in order to maintain her trade secret in it.

2. Draft a sample trade secret agreement.

See curriculum module Software Development and Licensing Contracts [Samuelson88b]

3. If Emily keeps a trade secret in the software, would she have to forego claiming a copyright in it?

It appears that one can maintain both a copyright and a trade secret in the same software product.

4. Could she register her program with the copyright office and still retain a trade secret? If so, how?

In registering her copyright, Emily will only be required to file the first and last 25 pages of source code. She can maintain information in the remainder of the program as a trade secret.

5. Could Emily maintain a trade secret in the software and still patent it? If so, how?

If Emily were able to obtain a patent for her program, disclosure would be required. Once something has been publicly disclosed, it can no longer be maintained as a trade secret. Note that this is consistent with the underlying policy of the patent law to grant exclusive rights for a period of time to the inventor in exchange for making his or her invention public.

6. Could Emily maintain a trade secret in the software if she licensed it to the government?

The FAR (civilian regulations) permit a developer to maintain a proprietary or trade secret interest in privately devel-

oped software. The DFARS (DoD regulations) do not specifically provide for the recognition of trade secrets, but do provide for protection of privately developed software under the restricted/limited rights provisions.

E. Licensing Existing Software to the Government.

1. Assume Emily licensed her copyrighted program to the government for use.

What rights would the government have in the software? In the documentation?

If Emily licensed her privately developed software to the government, the government would, under government procurement regulations, obtain restricted rights in the software. This means that, as to the machine-readable code, the government would, under both the civilian agency regulation (FAR) and Department of Defense regulation (DFARS), have minimum rights, including the rights to use the software with the computer for which it was acquired and use it with a backup if that computer becomes inoperative, to make backup copies, and to modify the software. These restricted rights in computer software tend to be site-specific. As to the related documentation, the government would receive the same restricted rights under the FAR, but would obtain limited rights under the DFARS. Limited rights tend to be government-wide rights and do not include a right to modify.

2. Could the government allow a software developer to use the program? To see the source code? To see the documentation?

Under the civilian agency regulations, the government would have the right to share the program, source code, or documentation with support contractors for the purposes of maintenance and enhancement. The DFARS do not claim such rights for the government. Often, however, the government does not obtain from the contractor access to the source code or proprietary documentation for fear, on the part of the developer, that such information might find its way into the commercial sphere.

3. Would the government still get only restricted rights in the software if Emily modified the program very slightly for the government?

If Emily modified her program slightly, after entering into government contract, the government would be in a position to claim the broader rights in the program if it was acquired under the DFARS. Slight modifications do not affect the restricted rights status of software under the FAR, however. Moreover, under the FAR, the contracting officer would have discretion to accept lesser rights in mixed funding situations, especially if the private contribution is 50% or more. Under the DFARS, a mixed funding alternative providing government purpose license rights in technical data (which includes software documentation) is available at the discretion of the government contracting officer. The DFARS provide extensive guidance as to when this alternative should be used. No mixed funding alternative is presently available for software.

F. Government Development Contract.

1. Now assume that Emily was preparing the soft-

ware for a government agency under a government contract. Could she copyright it?

In some circumstances, Emily would be able to copyright a program she developed for the government. This would depend, however, on the nature of the contract she was working under.

2. What would we need to know to determine if she could copyright it?

At present, the procurement regulations of both the civilian agencies (FAR) and the Department of Defense (DFARS) provide for situations under which a developer may claim a copyright in software and/or documentation developed under government contract. On the civilian side, however, the default setting is that the developer must get the contract officer's permission to assert copyright. Even then, the agency may, under the FAR, require that the developer assign the copyright to the government.

On the military side, on the other hand, the DFARS permits developers to claim a copyright in work performed under government contracts unless the "special works" clause of the DFARS has been included in the contract. If the "special works" clause is invoked, then either the work falls into the public domain or the government may attempt to claim the copyright directly under the "work made for hire" doctrine. It should be noted, however, that an attempt by the government to claim a copyright directly is probably invalid under §105 of the copyright law, which expressly prohibits the government from directly claiming a copyright in works produced under government contract. For the government's purposes, therefore, it is advisable to first have the developer claim the copyright and then assign it to the government under a standard assignment clause, as is done by the civilian agencies, if the government wishes to hold a copyright.

3. If Emily did copyright the program, how would this affect the government's rights in the program?

The government generally obtains what are called "unlimited rights" in software and documentation prepared under a government contract. This broad set of rights means that the government obtains, under the FAR, the rights to "use, disclose, reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, in any manner and for any purpose, and to have or permit others to do so" (§27.401 of FAR). Similarly, in contracts with the Department of Defense, unlimited rights give to the government a very broad license that includes the rights to "use, duplicate, release, or disclose, technical data or computer software in whole or in part, in any manner and for any purpose whatsoever, and to have or permit others to do so" (§227.471 of DFARS).

If Emily claims a copyright in the program, then the government's license is cut back in such a way that its rights extend only to use for government purposes, rather than for any purpose, as would otherwise be permitted under the unlimited rights provision. An important distinction between "unlimited rights" and "government purpose license rights" is that under the former, the government would have the right to use the software and documentation for commercial purposes or to allow others to do so, whereas under the

latter, it would have no right to commercially exploit the product, nor to allow another to do so.

4. Could the government modify the program if Emily has a copyright in it?

The government would have the right to modify the software, for any purpose, as part of its “unlimited rights” package. If Emily copyrights the software, the resulting “government purpose license” would limit modification to government purposes. Government purposes, as mentioned above, do not include the right to commercialize the software, nor the right to permit others to do so.

5. Could the government provide copies of the program to other developers for enhancement or maintenance work?

Both unlimited rights and government purpose license rights include the right to make the software available to other contractors for maintenance and enhancement purposes, since maintenance and enhancement is a legitimate government purpose.

Bibliographies

The bibliographies in this section contain references to books, articles, and cases related to intellectual property law and software. This literature is likely to be unfamiliar and confusing to computer professionals. To facilitate access to it, we have provided below background on the legal system and on legal research materials.

Notes on the U.S. Legal System

There are several sources from which intellectual property law is derived. Federal statutes, state statutes, court decisions interpreting and discussing statutes, and common law principles that have evolved over the years all have a part in forming the core of intellectual property law. In addition, there are federal regulations that affect intellectual property law and, in some instances, resemble a form of intellectual property law themselves.

Federal Law. There are three primary federal intellectual property statutes with which the reader may want to be familiar—the Copyright Act, the Patent Act, and the Semiconductor Chip Protection Act (SCPA). The Copyright Act and SCPA are found at 17 U.S.C. §101, *et seq.*, and §901, *et seq.*; the Patent Act is at 35 U.S.C. §1, *et seq.* These statutes govern the areas of patent, copyright, and chip protection and are the exclusive source of law for these forms of intellectual property protection. All litigation under such laws is conducted in federal courts.

Another federal statute dealing with intellectual property matters is the trademark law. Trademark law deals with rights in a particular name, logo, label, or other source-identifying mark. Formerly, trademark was exclusively state common-law doctrine, evolving through many years of court decisions. It now receives national treatment under a federal statute known as the Lanham Act, found at 15 U.S.C. §1151, *et seq.* Trademark litigation may proceed in federal or state court.

These federal statutes are interpreted and applied in cases arising within the federal court system. The federal court system consists of three levels: (1) the Supreme Court of the United States; (2) the United States Circuit Courts of Appeals (There are 13 such courts, referred to as “circuit courts.” Twelve of these courts serve regions of the country; the other

court deals with special subject matters.); and (3) the U.S. District Courts, which are trial-level courts within each state or within districts of a state.

Only decisions of the United States Supreme Court are the law throughout the country. That is, when the Supreme Court decides an issue, other courts are obliged to follow its precedent in subsequent cases dealing with similar issues. Decisions of the Courts of Appeals have similar precedential value, but only within their circuits (*i.e.*, only with respect to the circuit court itself and district courts within that circuit). The Supreme Court may give prior circuit court or district court decisions some weight in deciding an issue, but it is not required to do so.

Decisions at the district court level are not as strong with respect to controlling future decisions. Decisions—made at the trial level itself—may, however, serve as useful predictors of how another court will address a particular kind of issue or of how it will interpret a decision of the Supreme Court or a circuit court.

District court decisions can be appealed to the Court of Appeals for that particular circuit, and circuit court decisions can be appealed (generally through what is called a petition for *certiorari*) to the Supreme Court of the United States. The Supreme Court may accept (granting *certiorari*) or it may decline to review the case (designated as a denial of *certiorari*). A refusal to review a case in this way has no effect, one way or the other, on the validity of the decision, and it does not affect future decisions regarding the issue.

Decisions can be appealed either because the court below made an “error of law” (for example, incorrectly stating the legal standard) or, less commonly, because there was insufficient evidence to support a lower-court decision.

State Law. Another area of intellectual property law is trade secret law. Many intellectual property scholars would classify trade secret law as a form of *tort* (injury) law, rather than *property* (rights in an item) law. The tort approach to trade secret law views the injury as being an interference with a protected relationship (the confidential or “secret” relationship), rather than an infringement of rights in a property.

Trade secret law is exclusively controlled by state law, mostly through court decisions (common law),

rather than statutes. There are, therefore, variations in trade secret law from state to state. For a general understanding of trade secret law, see the 1939 *Restatement of Torts* [Restatement39]. Section 757 provides an excellent overview. See also section 2.3 of the module outline.

Federal Regulations. There is a set of regulations under which federal agencies procure software, the Federal Acquisition Regulation, or FAR. This main set of federal procurement regulations is supplemented by other provisions adopted by individual agencies (e.g., Department of Defense FAR Supplement and NASA FAR Supplement). These regulations, which allocate rights in software and technical data acquired by the government, set up a framework similar to a form of intellectual property law.

Interpretation of Laws. When there is a conflict in court decisions on a particular issue, the decision of the higher court controls. That is, Supreme Court decisions take precedence over circuit court and district court opinions, while circuit court decisions are controlling where in conflict with district court decisions. A circuit court decision is only binding within the circuit in which it was decided. On the other hand, Supreme Court decisions are binding everywhere. Sometimes a court may find the decision of another circuit persuasive if it is not in conflict with established precedent in its own circuit. Decisions of federal courts take precedence over state courts in interpreting the federal intellectual property statutes.

Where a conflict exists, federal statutes control over regulations. In addition, regulations must be consistent with the legislation permitting the issuing agency to adopt regulations.

In the state courts, the higher ranking appellate courts take precedence over lower courts in the state system. Decisions from other states have no precedential effect, although a state court may find an outside decision persuasive, if well reasoned. As to state law matters, such as trade secret law, state court decisions are of greater authority than are federal court decisions.

Understanding Legal Citations

The legal field has a set of prescribed citation forms for the various compilations of statutes and court decisions in which the record of the law is found. A summary of these citation forms follows. All legal citations in the bibliography are in the standard legal format. Note that not all minor variations on these basic forms will be discussed.

Federal Law Forms.

- Supreme Court Cases

<VOL. NO.> U.S. <PAGE NO.> (<DATE>), as in: *Diamond v. Diehr*, 450 U.S. 175 (1981).

The *United States Reports* are the official reports of United States Supreme Court decisions. These decisions are also available in the *Supreme Court Reporter* (S. Ct.) and *Supreme Court Reports, Lawyers' Edition, Second Series* (L. Ed. 2d).

- Courts of Appeals Cases

<VOL. NO.> F.2d <PAGE NO.> (<CIRCUIT NO.> Cir. <DATE>), as in: *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986).

The *Federal Reporter, Second Series* is the standard reference for U.S. Courts of Appeals decisions. In the example above, the case was heard by the U.S. Third Circuit Court of Appeals.

- District Court Cases

<VOL. NO.> F. Supp. <PAGE NO.> (<DISTRICT NAME> <DATE>), as in: *SAS Institute, Inc. v. S&H Computer Systems, Inc.*, 605 F. Supp. 816 (M.D. Tenn. 1985).

The *Federal Supplement* is the standard reference for U.S. District Court opinions. In the example above, the case was heard by the U.S. District Court for the Middle District of Tennessee.

- Federal Statutes

<TITLE NO.> U.S.C. §<SECTION NO.> (<DATE>), as in: Copyright Act of 1976, 17 U.S.C. §101, *et seq.* (1987).

The *United States Code* compiles the various statutes of the United States. It is organized into different "titles" (or parts) that collect together all the laws of a particular type. The Copyright Act is found in Title 17 of that code, the Patent Act in Title 35, the Semiconductor Chip Protection Act in Title 17, and the Trademark Act (Lanham Act) in Title 15.

- Federal Regulations

<VOL. NO.> C.F.R. §<SECTION NO.> (<DATE>), as

in: Federal Acquisition Regulation, 48 C.F.R. §1.000, *et seq.* (1987).

Regulations of federal agencies are compiled in the *Code of Federal Regulations*.

State Law Forms. There are also numerous collections of state statutes and court decisions that can be found in law libraries. State court decisions can be found in regional reporters—sets of volumes containing reported court cases—covering various regions of the country or in local state reporter systems. Local reporters are less often available for cases outside the state in which the law library is located. Statutes are generally found in state-specific codifications.

- Cases

<VOL. NO.> <LOCAL REPORTER> <PAGE NO.>, <VOL. NO.> <REGIONAL REPORTER> <PAGE NO.> (<DATE>), as in: *Analogic Corp. v. Data Translation, Inc.*, 371 Mass. 643, 358 N.E.2d 804 (1976).

Such a citation gives the location of the case in the local reporter (in this case, Massachusetts) and in the regional reporter (in this case, *North Eastern Reporter*, which includes cases from Massachusetts, as well as cases from other states in the region).

- Statutes

A typical citation of a state statute might be:

CAL. PENAL CODE §499c (West 1987).

This is a citation of a California statute dealing with trade secrets. “West” refers to the publisher of the compilation. This particular compilation of statutes, like many others, is annotated (*i.e.*, in addition to the language of the statute, the publisher has included information regarding the history of the law, cases interpreting it, and other related materials).

Availability of References

The books, articles, and cases listed in these bibliographies should be easily accessible in the law library of any law school, any county bar association library, and even the libraries of major law firms in your community. A few of the books may be available in general university libraries. Law libraries are usually organized so that the treatises on the law and other general books are located in one section of the library, law review volumes (for articles) in another (alphabetical by the name of the review, numerically by volume for each review), and cases and statutes

in still another section. Usually law libraries have maps on the wall to help users locate what they need.

Standard Treatises

There is, for each of the major subfields of intellectual property law, a standard scholarly treatise that is the most respected work in the field and provides the most complete guidance on the state of the law on particular issues. The standard treatises are listed below. (There is no standard treatise for federal acquisition regulations.)

Nimmer87

Nimmer, M. *Nimmer on Copyright, Vol. 1-5*. New York: Matthew Bender & Co., 1987.

Chisum87

Chisum, D. *Patents, Vol. 1-5*. New York: Matthew Bender & Co., 1987.

Milgrim87

Milgrim, R. *Trade Secrets, Vol. 1-2*. New York: Matthew Bender & Co., 1987.

McCarthy84

McCarthy, T. *Trademark Law, Vol. 1-2, 2nd Ed.* Rochester, N.Y.: Lawyer’s Co-Operative Publishing, 1984.

Stern86

Stern, R. *Semiconductor Chip Protection*. New York: Law & Business/Harcourt Brace Jovanovich, 1986.

Other Books on Intellectual Property Law

Most law libraries also have other books on intellectual property law, some written as overviews of the different types of intellectual property law, some more heavily focused on one subject (*e.g.*, copyright). Some of these books are in the standard narrative form. Others are “casebooks.” These teaching texts for law courses are topically organized. They contain selected excerpts from major cases on each topic, along with notes and questions raised by the case. Cases involving similar issues may also be mentioned.

Among the more respected intellectual property law books are the following:

Brown85

Brown, R., and R. Denicola. *Cases on Copyright, 4th Ed.* New York: Foundation Press, 1985.

A comprehensive casebook on copyright law, with excerpts from the major cases, as well as thought-provoking questions and comments on copyright issues and references to other cases and commentary. The book also gives extensive treatment to licensing and other contracting practices used in the entertainment industry. Unfair competition law, insofar as it provides additional protection to authors or entertainers, is also covered.

Choate87

Choate, R., W. Francis, and R. Collins. *Cases and Materials on Patent Law.* St. Paul: West Publishing Co., 1987.

A comprehensive, well-organized casebook on patent law, containing excerpts from major cases. The book also gives some attention to trade secret, trademark, and copyright issues, but its primary emphasis is on patent law.

Kaplan67

Kaplan, B. *An Unhurried View of Copyright.* New York: Columbia University Press, 1967.

A very readable history of copyright law and overview of its fundamental principles. Includes thoughtful predictions about its evolution.

Kitch86

Kitch, E., and H. Perlman. *Legal Regulation of the Competitive Process, 3rd Ed.* New York: Foundation Press, 1986.

This is as comprehensive a casebook on intellectual property law as currently exists, with relatively balanced treatment of copyright, patent, trade secret, and trademark law. The title of the book reveals the authors' orientation, which is to view intellectual property as a kind of regulation (*i.e.*, limitation by law) of competitive conduct.

Latman85

Latman, A., R. Gorman, and J. Ginsburg. *Copyright for the Eighties, 2nd Ed.* Charlottesville, Va.: Michie Company, 1985.

Another comprehensive casebook on copyright law, with case excerpts and commentary. It is entirely devoted to copyright law and gives in-depth treatment to many copyright issues either not covered by other books or treated only lightly.

Restatement39

Restatement of the Law, Torts. St. Paul: American Law Institute, 1939.

Restatements describe the state of the common law on topics of law. Each section of the *Restatement* talks about a particular legal issue. Section 757 of the *Restatement of Torts* (as this volume is generally called) concerns trade secret protection. This section and the commentary on it constitute the classic statement of what a trade secret is and is not and when trade secrets have been misappropriated.

Books on Computer Law

There is no dearth of books on “computer law” or “software legal protection.” But there are fewer good books on these subjects than one might wish. The problem is largely that the field of software law is a new one, and the law is as yet unsettled. Efforts to draw analogies between software legal issues and circumstances presented by prior cases have led to confusing results. Also, software as a technology is not generally well understood by lawyers and judges. Consequently, much that has been written on the topic is superficial or premature. Below are characterizations of the best sources now available, their scope, strengths, and focus.

Bender87

Bender, D. *Computer Law: Software Protection and Litigation, Vol. 1-2.* New York: Matthew Bender & Co., 1986.

This treatise provides a comprehensive discussion and analysis of patent, copyright, and trade secret laws as they relate specifically to software. The work opens with a discussion of the computer industry, which provides a context for how software protection issues arise. The chapters on copyright, trade secret, and patent law cover the prerequisites for each type of protection, the scope, the advantages, and the disadvantages. In addition to covering software protection offered by intellectual property law, the book discusses the contractual protection offered by software licenses. Treatment includes a brief overview of licensing principles, including source code escrow and shrink wraps. The appendices contain several relevant software protection articles, as well as Copyright Office regulations and some state statutes.

Davidson86a

Davidson, D., and J. Davidson. *Advanced Strategies for Buying and Selling Computers.* New York: John Wiley, 1986.

Although the intellectual property discussion is not quite mainstream material, this book offers a practical, in-depth discussion of the major legal, business, and tax decisions encountered by companies in the computer industry. The authors present strategies for making and implementing these decisions. The book presents a good overview of commercial law as it applies to software licenses, warranties, and limitations of liability. The section on protecting technology analyzes the current law from the perspective of a company aggressively seeking to protect proprietary technology, presents suggestions for structuring a software protection program, and discusses considerations in protecting and enforcing those rights. The chapter on employment agreements is specifically targeted to the software company employer and covers all aspects of such agreements. Acquisition of hardware and software is addressed from both the vendor's and the customer's perspective. The chapter on software licensing presents an overview of the issues that should be addressed in each license. The book concludes with a chapter on distribution arrangements and the legal issues, such as antitrust considerations, that can affect such arrangements.

Davis85

Davis, G. *Software Protection*. New York: Van Nostrand Reinhold, 1985.

This "user friendly" book is intended to give both practical and legal assistance to software developers, publishers, executives, and lawyers in the microcomputer industry, as well as to provide guidance to software users regarding their responsibilities under software licenses. In addition to providing a broad overview of intellectual property law as it relates to protecting software, the author identifies steps that can be taken to protect software without the daily services of a lawyer. The book covers copyright, trade secret, and patent protection, with a primary focus on copyright. Chapter 11, on software licenses, provides a useful analysis of the reason software is licensed and sold. The author, however, focuses only on object-code copyright licenses. Copyright registration, enforcement, and infringement are also discussed. Chapter 13 provides a useful discussion about licensing software to the government, but this chapter may soon become outdated, in light of forthcoming revisions to the regulations. This book is useful for its practical, in-depth coverage of copyright licensing.

Epstein84

Epstein, M. *Modern Intellectual Property*. New York: Law & Business/Harcourt Brace Jovanovich, 1984.

The major focus of this lawyer's guide to intellec-

tual property is trade secret law, but it also provides very brief coverage of copyright and patent law. The book has chapters on protection of ideas, restrictive covenants (promises not to work for competitors, and the like), and computer software. It has numerous appendices with some model forms and some relevant statutes.

Henry80

Henry, N., ed. *Copyright, Congress and Technology: The Public Record, Vol. V: CONTU's Final Report and Recommendations*. Phoenix: Oryx Press, 1980.

Contains the CONTU Report. (See [CONTU79].)

Hoffman87

Hoffman, P. *The Software Legal Book*. Croton-on-Hudson, N. Y.: Shafer Books, 1987.

A very basic "how-to" book with helpful practical suggestions about how to protect intellectual property rights in software.

Lautsch85

Lautsch, J. *American Standard Handbook of Software Law*. Reston, Va.: Reston Publishing, 1985.

This book was written as a user's manual to enable the software engineer to communicate with lawyers, tax advisors, and other professionals. It provides a broad survey of legal concepts, business implementation guidelines, and examples of documents such as publication and maintenance agreements and licenses. The book offers an overview of the rudiments of intellectual property, tort, and contract law as they affect the software developer. Additionally, software warranties and liability issues are covered. The book is unique in presenting legal principles in a context that can be understood by software engineers. It is recommended reading for students and teachers seeking a broad-based introduction to legal concepts affecting software. Although it does not contain extensive coverage of software licenses, it could serve as a useful teaching model for presenting legal concepts to the software engineering student who has no familiarity with the legal world.

Nash83

Nash, R. C., Jr., and L. Rawicz. *Patents and Technical Data*. New York: Government Contracts Program, George Washington University, 1983.

Although the federal and defense acquisition regulations affecting software have changed since this book was written, it is a very useful resource on the history of "data rights" regulations and is widely used by government attorneys.

Nimmer85

Nimmer, R. *The Law of Computer Technology*. Boston: Warren, Gorham & Lamont, 1985.

This work offers a comprehensive and sophisticated treatment of the legal issues spawned by the information age. Chapters 1 through 3 cover the law of patent, copyright, and trade secret. Chapter 4 provides an excellent discussion of the legal aspects of joint and sequential development of computer technology; it encompasses intellectual property, antitrust, and tax principles. The coverage of technology licensing in chapter 6 is a good overview of licensing law, spanning copyright, patent, and trade secret licenses, as well as software publishing and distribution contracts. The book also discusses computer-related torts, international trade considerations, computer crime, electronic publishing, and computer privacy. This book is highly recommended reading for teachers.

OTA86

Congress of the United States, Office of Technology Assessment. *Intellectual Property Rights in an Age of Electronics & Information*. Washington, D.C.: U.S. Government Printing Office, 1986.

Of all the books on software legal protection, this is the best and most interesting discussion of the challenges to traditional intellectual property goals presented by advances in electronic storage and transmission of information. If you want one book to use for a course on deep issues of intellectual property law, this is it.

Articles and Reports

Most of the articles below are law review articles and can be found in law libraries. This bibliography uses legal citation form for the the articles, which shows the volume number before the journal name, followed by the page citation and year of publication.

Bender86

Bender, D. "Protection of Computer Programs: The Copyright/Trade Secret Interface." 47 *University of Pittsburgh Law Review* 907 (1986).

Bender discusses numerous potential sources of conflict between copyright and trade secret protection for the same piece of software, and develops a theory to harmonize them.

Breyer70

Breyer, S. "The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, & Computer Programs." 84 *Harvard Law Review* 281 (1970).

This article questions the need for copyright protection for books and computer programs and hypothesizes how such works would be distributed in the absence of copyright.

Chisum86

Chisum, D. "The Patentability of Algorithms." 47 *University of Pittsburgh Law Review* 959 (1986).

Critical of Supreme Court decisions that have stated that algorithms are unpatentable ideas, Professor Chisum constructs an argument that protecting algorithms is consistent with patent doctrine and purposes underlying patent law.

Conley85

Conley, J., and R. Bryan. "A Unifying Theory for the Litigation of Computer Software Cases." 6 *Computer/Law Journal* 55 (1985).

This article advocates adoption of a copyright infringement standard focusing on the conduct of the alleged infringer (*e.g.*, did he or she make use of the plaintiff's code in developing his own?).

CONTU79

National Commission on New Technological Uses of Copyrighted Works. *Final Report*. 1979.

The "CONTU Report" Congress relied on in deciding to add computer programs to the copyright system. The entire report is reprinted in [Henry80].

Davidson83

Davidson, D. "Protecting Computer Software: A Comprehensive Analysis." 23 *Jurimetrics J.* 337 (1983).

The title says it all. Primary emphasis is on copyright law protection for software. The author has an expansive view of the scope of copyright for software.

Davidson86b

Davidson, D. "Common Law, Uncommon Software." 47 *University of Pittsburgh Law Review* 1037 (1986).

This article argues for common-law modification of copyright law to provide suitable protection for software and other information products. The "black box" test for infringement—*i.e.*, you can copy externals but not internals—is introduced here. An interesting theory, but not one used by judges.

Deasy88

Deasy, K., and A. Martin. "Seeking the Balance Between Government and Industry Interests in Software Acquisition." *14 Rutgers Computer and Technology Law Journal* 159 (1988). Also published as U.S. Air Force Electronic Systems Division Technical Report ESD-TR-87-114.

The authors recommend changes in the Department of Defense software acquisition regulations to achieve a better balance between government and industry interests. This is a report on consensus achieved as a result of a workshop in which government and industry representatives participated.

DisclosureNote86

"The Disclosure Requirements of 35 U.S.C. §112 and Software-Related Patent Applications: Debugging the System." *18 Connecticut Law Review* 855 (1986).

This article discusses the disclosure requirements of patent law as they affect software inventions. Quite useful discussion for those considering patent protection.

DSB87

Defense Science Board. *Report of the Defense Science Board Task Force on Military Software*. 1987.

The Defense Science Board, consisting of scientists who advise the Department of Defense on science-related issues, formed a task force to study how to improve the development and delivery of high-quality software to DoD. This report analyzes impediments to Department of Defense acquisition of advanced software systems. It discusses software "data rights" regulations as an impediment and recommends changes to the regulations.

Gilburne82

Gilburne, M., and R. Johnston. "Trade Secret Protection for Software Generally and In the Mass Market." *3 Computer/Law J.* 211 (1982).

Gilburne and Johnston outline the pitfalls and advantages of trade secret protection for software.

Goldstein86

Goldstein, P. "Infringement of Copyright in Computer Programs." *47 University of Pittsburgh Law Review* 1119 (1986).

The author articulates a theory that copyright protection for software should be quite narrow on account of its functional nature.

Grogan84

Grogan, A. "Decompilation and Disassembly: Undoing Software Protection." *1 Computer Lawyer* 1 (1984).

Grogan argues that making a copy of software in order to reverse-engineer it ought to be considered both an infringement of the copyright and a misappropriation of trade secrets.

HarvardNote82

"Toward a Unified Theory of Copyright Infringement for an Advanced Technological Era." *96 Harvard Law Review* 450 (1982).

It is argued that, in view of advances in copying technology, copyright infringement should be restricted to commercial and nearly identical copying.

Haynes87

Haynes, M., and S. Durant. "Patents and Copyrights in Computer Software Based Technology: Why Bother with Patents?" *4 Computer Lawyer* 1 (1987).

This article argues that it is worthwhile to patent software inventions because of the greater certainty under patent law, as compared to copyright law, of protection of structural features of software.

Karjala87

Karjala, D. "Copyright, Computer Software, and the New Protectionism." *28 Jurimetrics J.* 33 (1987).

This article asserts that copyright protection for computer programs should be limited to protection against outright copying of code. The article discusses the dangers of overprotection of software, especially with respect to the growth of the technology. Karjala is critical of many recent court decisions.

Kastenmeier85

Kastenmeier, R., and M. J. Remington. "The Semiconductor Chip Protection Act of 1984: A Swamp or Firm Ground?" *70 Minnesota Law Review* 417 (1985).

Thoughtful essay by a congressman and a congressional staffer, who develop guidelines for when legislative action might be appropriate in response to requests for creation of new forms of intellectual property protection. The SCPA experience is used as an illustration.

Kidwell85

Kidwell, J. "Software & Semiconductors: Why Are We Confused?" *70 Minnesota Law Review* 533 (1985).

Discusses the different kinds of confusion that software and semiconductors have created for legal policymakers and judges who attempt to make or interpret rules about intellectual property protection for software and semiconductors.

Kline86

Kline, M. "Requiring an Election of Protection for Patentable/Copyrightable Computer Programs." 6 *Computer/Law J.* 607 (1986).

Examines issues related to computer programs that meet the requisites for both copyright and patent protection. The author argues that the developer should be required to choose either copyright or patent protection, and should not be permitted to claim both.

Lange81

Lange, D. "Recognizing the Public Domain." 44 *Law & Contemporary Problems* 147 (1981).

Critical of some copyright and unfair competition cases that create property or property-like rights for things traditionally thought to be unprotectable, thereby threatening the public domain.

LoyolaNote87

"*Whelan Associates v. Jaslow Dental Laboratory*: Copyright Protection for the Structure and Sequence of Computer Programs." 21 *Loyola Los Angeles Law Review* 255 (1987).

Interesting overview and discussion of the *Whelan* decision.

Maier87

Maier, G. "Software Protection—Integrating Patent Copyright, and Trade Secret Law." 28 *Idea* 13 (1987).

Maier discusses how different aspects of software can be protected by different kinds of intellectual property rights.

Martin87

Martin, A., and K. Deasy. "Licensing of Intellectual Property Rights Needed for Software Support: A Life Cycle Approach." 28 *Jurimetrics J.* 223 (1988). Also published as U.S. Air Force Electronic Systems Division Technical Report ESD-TR-87-102.

The authors discuss the importance of making appropriate licensing arrangements for the maintenance and enhancement of software after delivery to the user.

Menell87

Menell, P. S. "Tailoring Legal Protection for Computer Software." 39 *Stanford Law Review* 1329 (1987).

An economic analysis of the need to "tailor" copyright protection for operating system software to enable compatible systems to be developed by competitors without the threat of infringement.

MinnesotaNote84

"Copyright Protection of Computer Programs: A Modification of the Substantial Similarity Test." 68 *Minnesota Law Review* 1264 (1984).

Argues that the "lay observer" test should not be employed in computer program copyright infringement cases because of its misleading character. Rather, infringement proceedings should focus on expert testimony regarding underlying program similarities. Argument is also made for broader rights to reuse components of computer programs to allow for incremental development of the technology.

Newell86

Newell, A. "The Models Are Broken, The Models Are Broken!" 47 *University of Pittsburgh Law Review* 1023 (1986).

Computer scientist's response to legal scholar Donald Chisum's article on the patentability of algorithms. While not disputing Chisum's conclusion that algorithms should be patentable under present law, Professor Newell argues that the theoretical models underlying the patent system are obsolete.

Nimmer86

Nimmer, R., and P. Krauthaus. "Copyright & Software Technology Infringement: Defining Third Party Development Rights." 62 *Indiana Law Review* 13 (1986).

The authors argue that because software is a technology and because technological progress is made through incremental improvements, copyright case law should distinguish between outright piracy of code and development of improved or expanded functionality.

Packard86

President's Blue Ribbon Commission on Defense Management. *A Quest for Excellence*. 1986.

Recommendations of a presidential commission known as the "Packard Commission," after its chairman David Packard, to the Department of Defense regarding acquisition of technical data and software. The report recommends a change in the

DFARS data rights policy so as to make it more balanced toward respecting the rights of private industry.

Patterson87

Patterson, L. R. "Free Speech, Copyright, & Fair Use." *40 Vanderbilt Law Review* 1 (1987).

Patterson argues that, under the U.S. Constitution, free speech considerations were intended to be incorporated into copyright law. To accomplish this, copyright liability should only be imposed on those who use the copyright competitively, and not on those who do so for noncommercial purposes.

PittNote86

"The Taking of Trade Secrets: What Constitutes Just Compensation?" *48 University of Pittsburgh Law Review* 247 (1986).

Discusses implications of *Monsanto* decision recognizing property interest in trade secrets. Examines alternatives for determining damages, based on infringement of property interest in trade secret.

Raskind85

Raskind, L. "Reverse Engineering, Unfair Competition & Fair Use." *70 Minnesota Law Review* 385 (1985).

Raskind discusses the history and implications of the reverse-engineering provision of the semiconductor chip law and the Congressional intent that, if reverse engineering is shown, "substantial identity," rather than merely "substantial similarity," must be shown in infringement actions.

Raskind86

Raskind, L. "The Uncertain Case for Special Legislation Protecting Computer Software." *47 University of Pittsburgh Law Review* 1131 (1986).

Recommending numerous changes in copyright law or interpretation to accommodate software, including a higher originality standard, a change in the modification provision, and a compulsory license feature.

Samuelson84

Samuelson, P. "CONTU Revisited: The Case Against Copyright Protection for Computer Programs." *1984 Duke Law Journal* 663 (1984).

Critical of the CONTU Report on which Congress relied in amending the copyright statute to add computer programs to its subject matter. Samuelson argues that copyright is not an appropriate form of protection for utilitarian subject matters that do not disclose their contents.

Samuelson85

Samuelson, P. "Creating a New Kind of Intellectual Property Law: Applying Lessons of the Chip Law to Computer Programs." *70 Minnesota Law Review* 471 (1985).

This article argues that the same reasons Congress had for creating *sui generis* legislation for semiconductor chip designs—namely, the utilitarian character of such designs—also apply to software. Therefore, it is appropriate to have *sui generis* legislation for software also.

Samuelson86a

Samuelson, P. "The Need for Reform of the Software Licensing Policy of the Department of Defense." *27 Jurimetrics J.* 9 (1986).

Discusses the software data rights regulations and policies of the Department of Defense and the interplay of these regulations and copyright law. Samuelson also discusses the interplay of these regulations and trade secret law, especially as to potential injunctive relief against disclosure of trade secrets.

Samuelson86b

Samuelson, P. "Allocating Ownership Rights in Computer Generated Works." *47 University of Pittsburgh Law Review* 1185 (1986).

Considers the problem of how intellectual property rights in output generated by a copyrighted computer program should be allocated. Samuelson concludes that among the potential beneficiaries—the machine, the user, the programmer, the user and programmer jointly, or no one—allocation of rights to the user is the most sensible policy and the most compatible with the purposes underlying the copyright law, except insofar as the output is a recognizable block of expression from the underlying program, in which case the programmer should have rights in it.

Samuelson87

Samuelson, P., K. Deasy, and A. Martin. "Proposal for A New "Rights in Software" Clause for Software Acquisitions by the Department of Defense." *4 Computer Lawyer* 32 (1987). Also published as U.S. Air Force Electronic Systems Division Technical Report ESD-TR-86-203.

Suggestions for changes to Department of Defense software acquisition policy. Problem with current acquisition regulations are identified, based on numerous interviews with government and industry representatives. Recommendations are aimed at achieving greater balance between government and industry interests.

Samuelson88a

Samuelson, P. “Modifying Copyrighted Software: Adjusting Copyright Doctrine to Accommodate a Technology.” *28 Jurimetrics J.* 179 (1988).

Samuelson argues that because software is a technology, there is need for wider rights to modify it—both by users and by third-party maintainers—than there is for traditional copyrighted works.

Samuelson88b

Samuelson, P., and A. C. Martin. *Software Development and Licensing Contracts*. Curriculum Module SEI-CM-15-1.0, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., April 1988.

This curriculum module gives an overview of legal issues involved in the development and distribution of software.

StanfordNote86

“Defining the Scope of Copyright Protection for Computer Software.” *38 Stanford Law Review* 497 (1986).

This note argues that structural similarities between programs should be a basis for copyright infringement and that screen displays should be considered part of the program’s copyrighted expression.

Stern85a

Stern, R. “Determining Liability for Infringement of Mask Work Rights Under the Semiconductor Chip Protection Act.” *70 Minnesota Law Review* 271 (1985).

Exhaustive survey of issues likely to arise in litigation over semiconductor chip designs under SCPA.

Stern85b

Stern, R. “Section 117 of the Copyright Act: Charter of the Software User’s Rights or an Illusory Promise?” *7 Western New England Law Review* 459 (1985).

Stern argues that certain cases have interpreted the special provision giving software users rights to make backup copies and modifications (§117) more narrowly than is sensible or than was intended by Congress.

Stern86

Stern, R. “The Bundle of Rights Suited to New Technology.” *47 University of Pittsburgh Law Review* 1229 (1986).

Stern proposes a new form of intellectual property

protection for non-code aspects of software, such as algorithms, instruction sets, programming languages, and input formats.

Sumner87

Sumner, J., and D. Plunkett. “Powerful New Software Protection in Europe: The Patent Trend Continues.” *4 Computer Lawyer* 1 (1987).

Discusses developments in international patent protection for software.

Cases

Relevant court cases are listed below. See “Understanding Legal Citations” above for a general discussion of the format used. Additionally, understanding of several terms and abbreviations is helpful for understanding the full meaning of these legal citations.

Relating to present and subsequent procedural history of a case:

- *cert. denied* — (*certiorari* denied): The court, generally the U.S. Supreme Court, exercised its discretion in deciding not to review a lower-court decision. A denial of *certiorari* has no effect on the precedential value of the lower court opinion.
- *aff’d* — (affirmed): An appellate court, after review, has affirmed, or agreed with, the decision of a lower court.
- *rev’d* — (reversed): An appellate court has disagreed with, and therefore reversed, a lower-court decision. An appellate court can reverse the lower-court decision in whole or in part. That part of a decision that has been reversed is no longer “good law.”
- *In re* — (in the matter of): Designation used to introduce a case in which only one party was involved.

Other abbreviations:

- U.S.P.Q. — (*U.S. Patent Quarterly*): A subject-matter-specific reporter publishing cases relating to patents, trademarks, and copyrights.
- C.C.P.A. — (Court of Customs and Patent Appeals): Predecessor (1929-1982) to the U.S. Court of Appeals for the Federal Circuit (Fed. Cir.).
- T.T.A.B. — (Trademark Trial and Appeal Board): An agency-level tribunal, within the Patent and Trademark Office, for resolving disputes related to trademarks.

Abele

In re Abele, 684 F.2d 902 (C.C.P.A. 1982).

Elaborating on the proper way to claim patent rights for software inventions.

Arndt

Williams v. Arndt, 626 F. Supp. 571 (D. Mass. 1985).

Infringement of copyright found as to program implementing system of stock purchasing described in a written work.

Arnstein

Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946).

Standard two-step analytic framework for copyright infringement. The first step involves expert testimony on the issue of whether the defendant made use of the infringing work. The second step excludes expert testimony and relies on a more intuitive judgement about whether piracy has occurred.

Artic

Midway Mfg. Co. v. Artic Int'l, Inc., 547 F. Supp. 999 (N.D. Ill. 1982), *aff'd*, 704 F.2d 1009 (7th Cir. 1983).

Manufacture and sale of circuit boards that speeded up the play of the plaintiff's copyrighted video game was held to be an infringing derivative work.

Atari

Atari v. JS&A Group, Inc., 597 F. Supp. 5 (N.D. Ill. 1983).

Contributory infringement of software copyrights was found where defendant sold device useful for duplication of video games made by the plaintiff.

Atlas

Atlas Powder Co. v. E.I. DuPont de Nemours & Co., Inc., 750 F.2d 1569 (Fed. Cir. 1984).

Discussing "equivalents" test for patent infringement.

Baker

Baker v. Selden, 101 U.S. 99 (1879).

Classic statement of the "idea/expression" dichotomy in copyright law. Copyright owner of a book about an accounting system could not prevent a second author from using very similar ledger sheets in his book on the same system.

Banner

Titanium Metals Corp. of America v. Banner, 778 F.2d 775 (Fed. Cir. 1985).

Patent may not issue on an old alloy, known to others through publication of an article, even though a new property of the alloy had been identified.

Beardsley

Continental Casualty Co. v. Beardsley, 253 F.2d 702 (2d Cir.), *cert. denied*, 358 U.S. 816 (1958).

Insurance form was copyrightable, but scope of copyright was very narrow because of its functionality. Virtual identity necessary to sustain infringement.

Benson

Gottschalk v. Benson, 409 U.S. 63 (1972).

Program for conversion of binary-coded decimal into pure binary was not patentable because a patent on it would be equivalent to a patent on the algorithm, and algorithms are unpatentable ideas.

BostonHockey

Boston Hockey Association, Inc. v. Dallas Cap & Emblem Mfg., Inc., 510 F.2d 1004 (5th Cir.), *cert. denied*, 432 U.S. 868 (1975).

Team won trademark infringement action against firm that made and sold patches with team trademark on them.

Bradley

In re Bradley, 600 F.2d 807 (C.C.P.A. 1979), *aff'd by an equally divided court, sub nom. Diamond v. Bradley*, 450 U.S. 381 (1981).

Microcode is patentable.

Broderbund

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

Similarities in screen display formats were the basis for a conclusion that Unison infringed the Broderbund program copyright.

Brulotte

Brulotte v. Thys Co., 379 U.S. 29 (1964).

Patent preemption of state contract law, which would have extended life of expired patent.

Cali

Cali v. Eastern Airlines, Inc., 442 F.2d 65 (2d Cir. 1971).

Discussing “experimental use” exception to patent’s novelty requirement.

Catalda

Alfred Bell & Co. v. Catalda Fine Arts Inc., 191 F.2d 99 (2d Cir. 1951).

Analyzing “originality” of mezzotints of “great master” paintings. To be “original,” a work must owe its origin to its author, not be copied from another, and be more than a trivial variation of preexisting works.

CES

CES Publishing Corp. v. St. Regis Publications Inc., 531 F.2d 11 (2d Cir. 1975).

“Consumer Electronics Monthly” is generic and not a trademark.

Christopher

E.I. DuPont de Nemours & Co., Inc. v. Christopher, 431 F.2d 1012 (5th Cir.), *cert. denied*, 400 U.S. 1024 (1970).

Trade secret misappropriation found where improper means used.

Colt

Christianson v. Colt Industries Operating Corp., 822 F.2d 1544 (Fed. Cir. 1987), *aff’d in part, rev’d in part*, 100 S.Ct. 811 (1988).

Responsibility of patentee to disclose elements of invention and not hold them as trade secrets.

Combustion

Combustion Engineering, Inc. v. Murray Tube Works, Inc., 222 U.S.P.Q. 239 (E.D. Tenn. 1984).

Reverse engineering from blueprints to make replacement boiler parts was not an infringement of copyright in the drawing.

ComputerStore

In re The Computer Store, 211 U.S.P.Q. 72 (T.T.A.B. 1981).

“The Computer Store” for computer sales outlet is generic, not a valid trademark.

Dann

Dann v. Johnston, 425 U.S. 219 (1976).

Software for bank cash deposit machine found to lack “invention,” even though it exhibited some capabilities not possessed by previous machines.

Diehr

Diamond v. Diehr, 450 U.S. 175 (1981).

Upholding software patentability where software was part of the rubber curing process. Diehr sought a process patent for a rubber curing process implemented by a computer program. The program permitted continuous monitoring and updating of information about temperature conditions within the rubber mold and signaled when the mold should be opened. The process produced perfectly cured rubber; prior to Diehr’s process, rubber was often improperly cured. This invention thus solved a long-standing industry problem. Because the process, considered as a whole, was performing a function of the sort that the patent laws were designed to protect (*e.g.*, transforming matter or reducing an article to a different state or thing), the Supreme Court held that Diehr had recited a patentable claim. This case has been hailed as a significant victory for the patentability of programs, even though it did not hold that a program in itself could be patentable as a process. Diehr is the most recent Supreme Court precedent on the patentability of computer programs. *Diehr* partly repudiates *Flook*, insofar as *Flook* suggested that software process claims had to be analyzed in terms of inventiveness of each constituent element of the process.

Dravo

Smith v. Dravo Corp., 203 F.2d 369 (7th Cir. 1953).

Trade secret misappropriation found where there was breach of confidential relationship created by negotiations.

Financial

Financial Information, Inc. v. Moody’s Investors Service Inc., 751 F.2d 501 (2d Cir. 1984).

No “originality” in compilation of information about municipal and corporate bonds; hence, no copyright protection.

Flook

Parker v. Flook, 437 U.S. 584 (1978).

It is important to understand that the *Flook* decision was made after *Benson*, but before *Diehr*. In *Flook*, the computer program at issue was not the whole of the process claimed to be patentable, but only the novel element of that process. For many years, those who operated catalytic converters had been measuring operating conditions such as temperature, pressure, and flow rates in order to calculate “alarm limits,” which indicated whether conditions within the converter were abnormal or dangerous, requiring corrective actions. Using a new algorithm, *Flook* had written a computer program that allowed the alarm limits to be continuously updated.

The Supreme Court thought Flook to be seeking, as Benson had, a patent on a mathematical formula, and hence rejected his claim as unpatentable. Although the Supreme Court did not say that processes involving computer programs were not patentable, its holding was discouraging for those who thought such protection desirable.

Formula

Apple Computer, Inc. v. Formula International, Inc., 594 F. Supp. 617 (C.D. Cal. 1984).

Competitor's transfer of copy of lawfully acquired Apple program to a different disk for repackaging for sale to customers infringed the Apple copyright; §117 defense was unavailing.

Franklin

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

Upholding the copyrightability of operating system programs over objections that they were utilitarian and therefore unprotectable processes.

Freeman

In re Freeman, 573 F.2d 1237 (C.C.P.A. 1978).

Two-step test for analyzing patentability of software inventions.

Frybarger

Frybarger v. IBM Corp., 812 F.2d 525 (9th Cir. 1987).

No infringement of video game copyright; similarities were in ideas and indispensable expression.

Gayler

Gayler v. Wilder, 51 U.S. 477 (1850).

Discussing the novelty requirements of patent law.

Ghiron

In re Ghiron, 442 F.2d 985 (C.C.P.A. 1971).

Detailed code need not be disclosed for software patent applications; flowcharts and block diagrams may be sufficient for disclosure of invention so long as people skilled in the computing field would be able to understand and implement the invention by writing a program from them

Gilliam

Gilliam v. American Broadcasting Cos., 538 F.2d 14 (2d Cir. 1976).

Copyright infringement was found where network deleted 27 percent of the text of Monty Python programs, more than the scope of the license allowed.

Gillman

Gillman v. Stern, 114 F.2d 28 (2d Cir. 1940).

Discussing the novelty requirements of patent law.

Gracen

Gracen v. Bradford Exchange, 698 F.2d 300 (7th Cir. 1983).

Higher standard of originality (a "substantial variation") necessary for derivative works.

Graham

Graham v. John Deere Co., 383 U.S. 1 (1966).

Classic statement of "invention" standard under U.S. patent law, and procedure to be followed in infringement decision in which the inventiveness of the patent is challenged.

Graver

Graver Tank & Mfg. Co. v. Linde Air Products Co., 339 U.S. 605 (1950).

"Equivalents" test for patent infringement.

Hubco

Hubco Data Products Corp. v. Management Assistance, Inc., 219 U.S.P.Q. 450 (D. Idaho 1983).

Making a copy of a computer program for reverse engineering purposes infringes the copyright.

INS

International News Service v. Associated Press, 248 U.S. 215 (1918).

Origins of "misappropriation" tort. INS was held liable for stealing verbatim AP news in uncopyrighted published newspapers. AP had "quasi-property" interest in its news, until the value of the news ceased.

Intel

Intel Corp. v. Radiation Inc., 184 U.S.P.Q. 54 (T.T.A.B. 1974).

"PROM" held to be a generic name, not a trademark.

Itoh

Digital Equipment Corp. v. C. Itoh & Co., 229 U.S.P.Q. 598 (D.N.J. 1985).

Layout of keys, set up of screen, and general ap-

pearance of video monitor was functional and not protectable as a trademark; no secondary meaning to other features for which trademark protection was sought.

Job's Daughters

Int'l Order of Job's Daughters v. Lindeburg and Co., 633 F.2d 912 (9th Cir. 1980).

In case involving jewelry manufacturer, the court held that use of the emblem of an organization did not constitute trademark infringement where emblem was used as functional part of jewelry design and no evidence was presented that customers had been misled into believing the organization sponsored or endorsed the jewelry.

Kalman

Kalman v. Kimberly-Clark Corp., 713 F.2d 760 (Fed. Cir. 1983).

Novelty "anticipation" doctrine of patent law.

Kenyon

Metallizing Engineering Co. v. Kenyon Bearing & Auto Parts Co., 153 F.2d 516 (2d Cir.), *cert. denied*, 328 U.S. 881 (1946).

Discussing the purpose of patent's novelty requirement.

Kewanee

Kewanee Oil Co. v. Bicrom Corp., 416 U.S. 470 (1974).

Discussing patent preemption and the relationship between patent and trade secret law.

Keystone

Keystone Bridge Co. v. Phoenix Iron Co., 95 U.S. 274 (1877).

The scope of patent rights is limited to "metes and bounds" of invention as described in the patent claims (*i.e.*, what they say the invention is).

Kramer

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

To prove copying of audiovisual work copyright in poker video game, similarities in underlying programs were admissible evidence.

Krofft

Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp., 562 F.2d 1157 (9th Cir. 1977).

Discussing analytic procedure to be used in copyright infringement cases.

Landsberg

Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485 (9th Cir. 1984).

Scope of copyright depends on the nature of the work. Narrow scope of copyright for more functional works, such as strategy book for games.

Lear

Lear, Inc. v. Adkins, 395 U.S. 653 (1969).

Patent preemption of state law.

Lotus1

Lotus Development Corp. v. Mosaic Software, — F. Supp. — (D. Mass.). Pending, No. 87-0074-K.

Copyright infringement claims based on similarities in the user interface and the screen displays for spreadsheet programs.

Lotus2

Lotus Development Corp. v. Paperback Software Int'l, — F. Supp. — (D. Mass.). Pending, No. 87-0076-K.

Copyright infringement claims based on similarities in the user interface and the screen displays for spreadsheet programs.

Manson

Brenner v. Manson, 383 U.S. 519 (1966).

Upholding Patent Commissioner's rejection of Manson's patent claims for a process to produce a particular chemical having no known utility, for failure to meet the "useful" requirement of §101 of the patent law.

Mazer

Mazer v. Stein, 347 U.S. 201 (1954).

A copyright in a statuette was challenged because it had been incorporated into a lamp as its base, which made it into part of a utilitarian work. Because statuette was not itself utilitarian, the copyright was upheld.

McGregor

McGregor-Doniger Inc. v. Drizzle Inc., 599 F.2d 1126 (2d Cir. 1979).

Trademark case setting forth the standard to be used to determine trademark infringement when goods are similar, but noncompeting.

MeadData

West Publishing Co. v. Mead Data Central, Inc., 799 F.2d 1219 (8th Cir. 1986).

Publisher of law books challenged computerized legal database service plans to put stars and page citations into the database. Defendant loses challenge to the “originality” of West page citations. Preliminary injunction issued.

Megapulse

Megapulse, Inc. v. Lewis, 672 F.2d 959 (D.C. Cir. 1982).

Contractor could get injunction to prevent government from releasing trade secret data for competitive procurement.

MerrillLynch

Paine, Webber, Jackson & Curtis, Inc. v. Merrill Lynch, Pierce, Fenner & Smith, Inc., 564 F. Supp. 1358 (D. Del. 1983).

Rejecting argument that data processing inventions are unpatentable. Case was subsequently settled, so there has been no appellate review of the holding.

Meyer

In re Meyer, 688 F.2d 789 (C.C.P.A. 1982).

Denying patent to medical expert system because it only mechanized mental process.

Monsanto

Ruckelshaus v. Monsanto Co., 467 U.S. 986 (1984).

United States Supreme Court case recognizing a property interest in trade secrets.

Morse

O'Reilly v. Morse, 56 U.S. 62 (1854).

Morse claimed a patent on all devices that used electromagnetic force to communicate signals over long distances, even though he had only invented one device to do this. His claim was likened to a claim for a patent on scientific ideas. Scope of patent restricted to that which patentee invented. No right to derivative inventions.

Morton-Norwich

In re Morton-Norwich Products, Inc., 671 F.2d 1332 (C.C.P.A. 1982).

Discussing whether shape of container was functional and hence disqualified for trademark protection; because other shapes could be used, court decided shape was not functional.

MotionPicture

Motion Picture Patents Co. v. Universal Film Mfg. Co., 243 U.S. 502 (1917).

Patentee cannot unlawfully extend patent protection through licensing.

NEC

NEC Corp. v. Intel Corp., — F. Supp. — (N.D. Cal. 1989). Case not yet published.

Holding that microcode is copyrightable but that NEC had not infringed Intel's copyright because Intel failed to monitor copyright notices and because code was too dissimilar to be infringing.

Nichols

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930).

“Pattern of abstraction” test for copyright infringement discussed in case involving two plays with similar plots.

Paula

C.M. Paula Co. v. Logan, 355 F. Supp. 189 (N.D. Tex. 1973).

No copyright infringement where purchaser of copy of copyrighted work incorporated it into product that was then sold to the public.

Pennwalt

Pennwalt Corp. v. Durand-Wayland, Inc., 833 F.2d 931 (Fed. Cir. 1987).

Software version of hardware invention for fruit sorting was not infringing because components were not equivalent on an element-by-element basis.

Pitt

Univ. of Pittsburgh v. Champion Prods., Inc., 686 F.2d 1040 (3d Cir.), *cert. denied*, 459 U.S. 1087 (1982).

Discusses issues related to use of team emblem on products to capitalize on public desire to identify with team. Court permitted University of Pittsburgh to seek injunction against future use of trademark by defendant.

PlainsCotton

Plains Cotton Cooperative Assn. v. Goodpasture Computer Service, Inc., 807 F.2d 1256 (5th Cir. 1987).

Like *Whelan*, *Plains Cotton* involved charges of copyright infringement based solely on structural

similarities between software programs. The Fifth Circuit reached a conclusion different from that reached by the Third Circuit in the *Whelan* case. Plains Cotton claimed that Goodpasture had obtained software design specifications by hiring four former Plains Cotton employees who had developed Plains Cotton's mainframe program and who had done the design specifications for a PC version of it. Though the appellate court agreed that there were significant organizational similarities between the two programs, it concluded that market factors played an important role in determining the sequence and organization of cotton marketing software. Such organizational similarities were, therefore, unprotectable "ideas" of the software, not "expression." The court in *Plains Cotton* expressly declined to follow the Third Circuit's *Whelan* decision or apply its test for software copyright infringement. It did not even cite *SAS*. It applauded and relied upon the reasoning in *Synercom*.

Polaroid

Polaroid Corp. v. Polarad Electronics Corp., 287 F.2d 492 (2d Cir.), cert. denied, 368 U.S. 820 (1961).

No infringement of trademark was found, despite similarity in name, where parties' goods were different and plaintiff delayed in bringing action.

Q-Co.

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

Former employee's competitive program did not infringe copyright, despite similar terms in new screens and modular structure of the competitive program because no "expression" was taken from the employer's program.

Roth

Roth Greeting Cards v. United Card Co., 429 F.2d 1106 (9th Cir. 1970).

Case which originated the "look and feel" test for copyright infringement cases.

Salkeld

Universal Athletic Sales Co. v. Salkeld, 511 F.2d 904 (3d Cir. 1975).

Application of *Baker v. Selden* to copyrighted exercise charts. Chart similarities as to exercises not a basis for infringement; graphic depictions of the exercises were different.

SAS

SAS Institute, Inc. v. S&H Computer Systems, Inc., 605 F. Supp. 816 (M.D. Tenn. 1985).

In addition to taking 44 lines out of 186,000 lines of code, structural similarities between the programs were the basis of infringement of copyright.

Sears

Sears, Roebuck & Co. v. Stiffel Co., 376 U.S. 225 (1964).

Federal patent law preempts state common law when it gives innovators equal or greater protection than patent or copyright law would provide.

Sheldon

Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936).

Similarities in plot structures of two dramatic works (incident-by-incident sequence within several scenes) was the basis for copyright infringement.

Sherwood

In re Sherwood, 613 F.2d 809 (C.C.P.A. 1980).

Discussing the "disclosure" and "best mode" requirement of patent law.

Smith

In re Smith, 714 F.2d 1127 (Fed. Cir. 1983).

Discussing "experimental use" of patent's novelty requirement.

SnowCrest

Coca-Cola Co. v. Snow Crest Beverages, Inc., 64 F. Supp. 980 (D. Mass. 1946).

Trademark case discussing contributory infringement. One who supplies goods knowing the customer will pass them off as those of the trademark owner, or one who induces customers to buy with the intention of passing them off as those of the trademark owner, can be held liable for contributory infringement.

Softklone

Digital Communications Assocs., Inc. v. Softklone Distributing Corp., 659 F. Supp. 449 (N.D. Ga. 1987).

Main menu screen of popular "Crosstalk" communications program was held not to be protected by the copyright in the underlying computer program. However, DCA had a separate copyright in a compilation of program terms, which was reflected in the main menu screen. Softklone's similar, but not

identical, menu screen in its competitive program was held to infringe DCA's compilation copyright. Softklone's use of DCA's distinctive capitalization and highlighting of the first two letters of the commands in the compilation and the execution of commands through typing of the first two letters were part of DCA's protected expression.

Sony

Sony Corp. v. Universal City Studios, 464 U.S. 417 (1984).

This case involved a copyright infringement charge against a manufacturer of a videotape machine that could be used to copy copyrighted movies from television. No infringement was found because there were substantial noninfringing uses of the machines; any copying of copyrighted works for time shifting purposes in home was fair use.

Straus

Bobbs-Merrill Co. v. Straus, 210 U.S. 339 (1908).

Attempt to restrict a purchaser's right to resell a copy of a copyrighted work was held to be outside the scope of exclusive rights provided by copyright law.

Strohon

Midway Manufacturing Co. v. Strohon, 564 F. Supp. 741 (N.D. Ill. 1983).

Court found infringement of video game program copyright but not of audiovisual copyright.

Synercom

Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978).

Synercom was the first software copyright case to consider whether structural similarities in the sequence and ordering of input formats could give rise to copyright liability. Synercom had developed a structural analysis program for engineers. Input formats for this program were described in Synercom's user manual for the program. Engineering Dynamics, Inc. (EDI) used these input formats in its own preprocessor program, which University Computing Co. commercially distributed. There was evidence in the case that there were hundreds of structural analysis programs available on the market and that only Synercom's and EDI's used the same formats. Moreover, it was clear that EDI had adopted these formats in order to compete more effectively with Synercom's program.

EDI raised two defenses to Synercom's input format infringement claim. First, relying on a set of cases that had ruled that blank forms were not proper subject matter for copyright (because blank

forms do not themselves communicate information but only serve as receptacles for information supplied by users), EDI contended that Synercom's input formats were uncopyrightable. The court rejected this claim, saying that the input formats expressed to the user the sequencing of data for simplified access to the computer programs. Because of these expressive qualities, Synercom's input formats were copyrightable subject matter.

Though Synercom prevailed as to EDI's first defense, it lost on the second defense, which was that EDI had taken only the "idea" of the formats, or more precisely, that EDI had taken no more "expression" of Synercom's formats than was necessary to take the "idea." EDI relied on a series of copyright cases that had ruled that when the "ideas" in a work were inextricably interconnected with the "expression," it was acceptable for others to take the expression in order to be able to take the idea.

EDI convinced the judge that Synercom's input formats were analogous to the "H" pattern for car stick shifts. This pattern may have been chosen randomly from a number of alternatives. It may be capable of being expressed in a variety of ways (such as a prose description, a diagram, or a photograph, each of which might be copyrightable). Yet that would not mean that the first car manufacturer to copyright a diagram of the "H" pattern could prevent other car manufacturers from using this pattern in their own cars or from using diagrams of a stick shift pattern. The "H" pattern would be the unprotectable "idea" depicted in the diagram. Use of this pattern by other manufacturers would be socially desirable, as it would reduce the retraining of drivers. Being unable to discern what "idea" there might be in Synercom's input formats, if not in their order and sequence, the judge decided that EDI had taken only the "idea" in Synercom's work, and so EDI had not infringed the copyright.

Taylor

Taylor Instrument Cos. v. Fawley-Brost Co., 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944).

Claim of copyright in circular chart used in conjunction with a temperature recording device was invalid because chart had been part of a patented device whose patent had expired.

TechnicalPublishing

Technical Publishing Co. v. Lebar-Friedman Inc., 729 F.2d 1136 (7th Cir. 1984).

"Software News" for magazine is generic and not a trademark.

Uniden

E.F. Johnson Co. v. Uniden Corp. of America, 623 F. Supp. 1485 (D. Minn. 1985).

Software copyright infringement was found because the defendant copied the detailed implementation of a portion of another firm's program.

Vault

Vault Corp. v. Quaid Software Ltd., 655 F. Supp. 750 (E.D. La. 1987).

Refusing to enforce "shrink wrap" license restrictions on purchaser copying, reverse engineering, and modifications, despite a Louisiana statute making "shrink wrap" licenses enforceable, because under federal copyright law, purchasers had rights to do such things.

Videotronics

Videotronics Inc. v. Bend Electronics, 564 F. Supp. 1471 (D. Nev. 1983).

Rejecting trade secret claim involving a copyrighted video game.

Walter

In re Walter, 618 F.2d 758 (C.C.P.A. 1980).

Elaborating on *Freeman* test for analyzing the patentability of software inventions.

Wexler

Wexler v. Greenberg, 399 Pa. 569, 160 A.2d 430 (1960).

Discussing former employee's responsibilities as to former employer's trade secrets.

Whelan

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 93 L. Ed. 2d 831 (1987).

Jaslow hired Whelan to develop a program to help him manage his dental laboratory business. He paid for the development and gave Whelan extensive access to his business to enable her to learn what functions needed to be included in such a program. Whelan wrote the "Dentalab" program in EDL to run on IBM mainframes. She delivered it to Jaslow and then arranged for commercial distribution to other dental labs. Jaslow decided that there would be a good market for software of this sort written in BASIC for IBM personal computers. Having taught himself to program, Jaslow wrote and then began to market such a PC program. Whelan sued.

At the trial court level, *Whelan* did not seem to be a

"structure" case at all. The trial judge seemed to be more impressed with how the two programs performed, speaking of similarities in the manner in which the two programs operated, controlled, and regulated the flow of information to the computer, as a basis for finding Jaslow guilty of infringing Whelan's copyright. The trial judge was also impressed by similarities in screen displays, though without clarifying whether he considered this to be a part of the program's expression.

On appeal, the Third Circuit upheld the lower court's ruling. The Third Circuit, however, characterized the similarities between the Whelan and Jaslow dental lab business software as "structural" similarities. The appellate court recognized that Jaslow's program was not a line-by-line translation of the Whelan program, that Jaslow's program was written in a different programming style, used different algorithms, and had many structural features dissimilar to Whelan's program. Yet some data and file structures were similar, and five subroutines performed the same function; these, the appellate court decided, were enough to show infringement of the copyright. Screen display similarities confirmed for the appellate court that the underlying programs had substantially similar structures.

The Third Circuit's analysis in the *Whelan* decision was very lengthy and complex. It may suffice, however, to say that the court relied upon cases in which the structural aspects of novels and plays had been the basis of the infringement determination, and upon the compilation provisions of the copyright law that provide that sequence and ordering of data can be copyrighted. Partially distinguishing *Synercom* on the grounds that the input formats in that case were structurally simpler than full programs, the appellate judges in *Whelan* also took issue with the other court's conclusion. They found another answer to the *Synercom* conundrum: if sequence and order was the expression, what was the separate idea being expressed? The judges in *Whelan* said that the "idea" was the general purpose or function of the program, and the approach chosen by a first software author was his or her "expression." Because there were other dental lab programs on the market that had different file and data structures from Whelan's, Jaslow could not claim he was only copying ideas. This case is very controversial, its reasoning much disputed, and its conclusion probably wrong, in view of the copyright law's exclusion from protection of processes and procedures.

White

White Consolidated Industries, Inc. v. Vega Servo-Control, Inc., 713 F.2d 788 (Fed. Cir. 1983).

Discussing the "disclosure" requirement in a software case.

Wilbur-Ellis

Wilbur-Ellis Co. v. Kuther, 377 U.S. 422 (1964).

Patentee had no right to control purchaser's adaptation of the patented machine; no exclusive right of the patentee was violated.

Wilkins

Williams & Wilkins Co. v. United States, 487 F.2d 1345 (Ct. Cl. 1973), *aff'd by an equally divided court*, 420 U.S. 376 (1975).

"Fair use" defense was upheld for a library in suit by publisher based on the library's policy of making copies of journal articles for medical researchers.

Williams

Williams Electronics, Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982).

Images depicted in "attract mode" of audiovisual game are "fixed in a tangible medium of expression," despite changing nature of images. Copyright can, therefore, be claimed in images, as well as in the underlying program.

Yardley

In re Yardley, 493 F.2d 1389 (C.C.P.A. 1974).

Discussing the relationship of copyright and design patent protection.