Catholic University Journal of Law and Technology

Volume 26 | Issue 2 Article 6

2018

Providing Protection to Programmers' Works: Disregard the Merger Doctrine and Adopt the Application Approach

Akshay Jain
Catholic University of America (Student)

Providing Protection to Programmers' Works: Disregard the Merger Doctrine and Adopt the Application Approach

Akshay Jain[+]

A programmer develops an application program that provides users information about any object by capturing a picture of it.[1] The program would provide the name of the object and other factual information.[2] The copyrightable aspects of the programmer's work has copyright protection from the time of creation.[3] After distributing, copying, and creating derivatives of the program in the market for some time, he or she learns someone else has been selling a program with

[+] J.D. Candidate, May 2019, The Catholic University of America, Columbus School of Law, B.A. 2012; University of California - Riverside. I would like to thank Professor Susanna Frederick Fischer for her assistance in the research and writing of this comment. I would also like to extend a special thanks to my family and friends for their contributions and support. Finally, I would like to thank the hard work and dedication of the associates and the editors of the Journal of Law and Technology on the preparation of this comment.

1 The inquiry in this comment is not limited to applications for phones but also includes computers. Osas Obaiza, *This "Search by Photo" App for iPhone Blows Google Goggles Out of the Water*, IOS.GADGETHACKS.COM (Mar. 21, 2014, 12:27 PM), https://ios.gadgethacks.com/how–to/search–by–photo–app–for–iphone–blows–google–goggles–out–water–0151197/.

2 In order for the computer to produce the desired results, the programmer must engage in programming, which includes defining, developing, writing, testing, documenting, and maintaining the program, in order for the computer to produce the desired results. G. MICHAEL SCHNEIDER, STEVEN W. WEINGART & DAVID M. PERLMAN, AN INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING WITH PASCAL 2–5, 106 (John Wiley & Sons 2d ed. 1982); *see, e.g.*, David R. Pakarek Krohn, *Media–Rich Input Application Liability*, 17 MICH. TELECOMM. TECH. L. REV. 201, 206 (2010) (explaining how Shazam is a smart-phone application that provides users information about a song by capturing a snippet of a track).

3 17 U.S.C. § 102(a) (2012); U.S. COPYRIGHT OFFICE, COMPENDIUM OF U.S. COPYRIGHT OFFICE PRACTICES § 102.2(A) (3d ed. 2017); *see, e.g.*, Eldred v. Ashcroft, 537 U.S. 186, 195 (2003) (finding Congress intended to provide federal copyright protections starting with the creation of the work, not publication).

similar functions.[4] The programmer files a complaint against the alleged infringer – the person who he or she believes stole his or her work.[5] However, the judge dismisses the claim because (a) the aspects of his or her original program that were allegedly infringed were not protected by the Copyright Act of 1976 ("the Copyright Act")[6] and (b) he or she failed to register his or her program with the United States Copyright Office before filing suit.[7]

The Copyright Act does not provide sufficient protection to a program for a programmer to receive a legal remedy for copyright infringement because when the function merges with the process,[8] courts will not provide protection to the process of program.[9] Applying the merger doctrine defeats the purpose of creating a program and the Congressional intent of the copyright law because the

---

4 17 U.S.C. § 106 (2012) (stating "[t]he owner of copyright under this title has the exclusive rights to do and to authorize any of the following: (1) to reproduce the copyrighted work . . . ; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies . . . of the copyrighted work to the public . . . ; [and] (5) to display the copyrighted work publicly."); *see, e.g.*, Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655, 659 (4th Cir. 1993) (reproducing, preparing derivatives, and distributing the plaintiff's computer program was improper because the program was similar to the plaintiff's program).

5 17 U.S.C. § 501 (2006); *see also* Julian Velasco, *The Copyrightability of Nonliteral Elements of Computer Programs*, 94 COLUM. L. REV. 242, 269–70 (1994) (applying the abstraction–filtration–comparison ("AFC") method, a test which involves distilling the program to determine whether the nonliteral aspects of the program were substantially similar between the two works, to a claim of copyright infringement of a computer program); *see, e.g.*, Softel, Inc. v. Dragon Med. & Sci. Communs., 118 F.3d 955, 963 (2d Cir. 1997) (stating that the defendant was sued for producing similar programs with those capabilities and the court applied the AFC test).

6 17 U.S.C. § 102 (2010); *see, e.g.*, Comput. Assocs. Int'l Inc. v. Altai, Inc., 982 F.2d 693, 706, 715 (2d Cir. 1992) (explaining how the plaintiff's program had no copyright protection based on the AFC test).

7 17 U.S.C. § 411(a) (2010); *see, e.g.*, Fourth Estate Pub. Benefit Corp. v. Wall-Street.com, LLC, 856 F.3d 1338, 1341 (11th Cir. 2017) (explaining the dismissal of the copyright infringement suit because the plaintiff's work was not registered and that the plaintiff should have waited until the Copyright Office notified the plaintiff about the result of the application).

8 The main function of an application is unprotected, but the process to create the main purpose is usually protected. Cognotec Servs. v. Morgan Guar. Tr., Co., 862 F. Supp. 45, 49 (S.D.N.Y. 1994); *see* 17 U.S.C. §§ 102(b), 504 (2012) (excluding "process" from copyright protection and providing remedies for infringement); *see also* Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1345–46 (5th Cir. 1994) (stating certain parts of the process, such as that the writing language of an application is protected).

9 *See* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 836 (10th Cir. 1993) (demonstrating how the source and object code of a program are not protected when doctrines of merger and scenes a faire apply); Joseph G. Arenault, *Software Without Source Code: Can Software Produced by a Computer Aided Software Engineering Tool Be Protected*, 5 ALB. L.J. SCI. & TECH. 131, 161 (1994) (arguing how copyright law intended to protect the source code and derivatives against "verbatim copying;" therefore, the program as a whole must be protected); *see, e.g.*, Aymes v. Bonelli, 47 F.3d 23, 27 (2d Cir. 1995) (holding the defendant

programmer created a program to distribute to the public and to promote innovation.[10] The copyright law "encourage[s] individual effort and creativity by granting valuable enforceable rights."[11]

Congress must view the copyright scope of a program from the programmer's perspective, understanding the artistic process to create a program, such as writing algorithms.[12] Furthermore, computer programs differ from other types of literary works because the nature and design of programs are highly complex.[13]

---

could copy the plaintiff's program because the modifications the defendant made to the plaintiff's program made it the defendant's work).

10 The merger doctrine states, "when there are a limited number of ways to express an idea, the idea is said to 'merge' with its expression, and the expression becomes unprotected." Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1359 (Fed. Cir. 2014); *see also* John Swinson, *Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection*, 5 HARV. J.L. & TECH. 145, 181 (1991) (explaining how "the expression and idea have merged where there are a few or no ways of expressing a particular idea.").

11 Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 620 (7th Cir. 1982), *superseded by statute* FED. R. CIV. P. § 52(a) (2017); Andrew O. Martyniuk, *Abstraction-Filtration-Comparison Analysis and the Narrowing Scope of Copyright Protection for Computer Programs*, 63 U. CIN. L. REV. 1333, 1368–69 (1995); *see* S. REP. NO. 94-473, at 51 (1975) (illustrating how "[a]uthors are continually finding new ways of expressing themselves, but it is impossible to foresee the forms that these new expressive methods will take."); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977, 1053–55 (1933) (explaining how technology is rapidly changing and, as a consequence, is challenging the scope of copyright law). *See generally* John Shaeffer, *Software as Text*, 33 SANTA CLARA COMPUT. & HIGH TECH. L.J. 324, 343, 346 (2017) (describing the initial purpose of a program is to be functional and be shared among other programmers, but this purpose is prevented through the protection of the original programmer's coding through copyright laws).

12 *See* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1499 (D. Minn. 1985) (holding the plaintiff had "contributed something 'recognizably its own' to the compilation of standardized programming techniques."); JAMES V. VERGARI & VIRGINIA V. SHUE, FUNDAMENTALS OF COMPUTER-HIGH TECHNOLOGY LAW 511 (American Law Institute-American Bar Association Committee on Continuing Professional Education 1991) (stating that understanding the copyright scope of computer programs depends on learning how they are made from the viewpoint of the programmer); Donald F. McGahn II, *Copyright Infringement of Protection Computer Software: An Analytical Method to Determine Substantial Similarity*, 21 RUTGERS COMPUTER & TECH. L.J. 88, 132 (1995). An algorithm is a "series of step-by-step instructions leading to a specific result." Christopher M. Mislow, *Computer Microcode: Testing the Limits of Software Copyrightability*, 65 B.U.L. REV. 733, 741 (1985); *see also* J.M. YOHE, AN OVERVIEW OF PROGRAMMING PRACTICES 222 (Association for Computing Machinery, Inc. 1974) (programming entails not only writing the program but also developing the idea for the program).

13 David M. Maiorana, *Privileged Use: Has Judge Boudin Suggested a Viable Means of Copyright Protection for the Non-Literal Aspects of Computer Software in Lotus Development Corp. v. Borland International?*, 46 AM. U.L. REV. 149, 151 (1996); Shaeffer, *supra* note 11, at 410 (comparing and contrasting different types of literary works with the complexity of programs); *see also* VERGARI & SHUE, *supra* note 12, at 1 (describing how "the relationship between technology, society, and the law are becoming important to every segment of society" that technology transcends our society and has far-reaching implications on our everyday life). *See generally* 17 U.S.C. § 101, 102(a) (2010) (defining literary works and what type of

Congress must amend the Copyright Act by abrogating the judicially created concept, the merger doctrine, and not apply it to programs.[14] The intent of the copyright law would align with the public understanding the value of programs, and programmers would have an incentive to create new programs because their program would have copyright protection.[15]

Once the Copyright Act provides the programmer with sufficient protection, he or she may have to enforce his or her legal rights by filing a lawsuit.[16] To sue for copyright infringement, the programmer has to register his or her program.[17] However, when the Court considers the work was registered, the approaches among the circuits are split.[18] Courts adopt either the registration[19] or application[20] approach.[21]

Congress must adopt the application approach because programmers should

---

protection literary works receive).

14 *See* Swinson, *supra* note 10 (stating the merger doctrine is an "unhelpful" doctrine to determine whether the idea of the computer program can be expressed in many different ways); *see also* Baker v. Selden, 101 U.S. 99, 102, 105 (1879), *abrogated by* 17 U.S.C. § 102 (2010) (describing the idea of expanding the idea-expression line to include the merger doctrine). The amendment to the Copyright Act would include other types of computer programs.

15 17 U.S.C. § 103 (2010); *see also* Peter D. Aufrichtig, *Copyright Protection for Computer Programs in Read Only Memory Chips*, 32 COPYRIGHT L. SYMP. 133, 172 (1982) (explaining how the copyright law must "provide adequate protection for proprietary software" by protecting "more than just direct copying.").

16 *See* 17 U.S.C. § 411 (2010) (describing the procedures to file a law suit for copyright infringement); Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1059 (1989); *see, e.g.*, Brooks-Ngwenya v. Indianapolis Pub. Schs., 564 F.3d 804, 805, 807 (7th Cir. 2009) (stating that the plaintiff sued the defendant for copyright infringement and must register her program before suit).

17 17 U.S.C. § 411(a) (2010).

18 *Compare* Fourth Est. Pub. Benefit Corp. v. Wall–Street.com, LLC, 856 F.3d 1338, 1340 (11th Cir. 2017) (stating registration was accepted upon receiving approval or denial), *with* Apple Barrel Prods. v. Beard, 730 F.2d 384, 386–87 (5th Cir. 1984) (stating the registration requirement can be met by filing the application).

19 Under the registration approach, the plaintiff meets the requirements of 17 U.S.C. § 411(a) upon receipt of the certificate of registration or the Copyright Office approving or rejecting the application. Mays & Assocs. v. Euler, 370 F. Supp. 2d 362, 368 (D. Md. 2005); *see, e.g.*, Kernel Records Oy v. Mosley, 694 F.3d 1294, 1302 (11th Cir. 2012) (holding since the plaintiff did not register its device before filing a lawsuit, it could not bring claim for copyright infringement).

20 Under the application approach, the plaintiff meets the requirements of 17 U.S.C. § 411(a) upon filing the application for registration with the Copyright Office. Lakedreams v. Taylor, 932 F.2d 1103, 1108 (5th Cir. 1991); *see, e.g.*, Chicago Bd. Of Educ. v. Substance, Inc., 354 F.3d 624, 631 (7th Cir. 2003) (stating that the registration of the work occurred because the Copyright Office received the plaintiff's application).

21 La Resolana Architects v. Clay Realtors Angle Fire, 416 F.3d 1195, 1202 (10th Cir. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010); Matthew J. Astle, *Help! I've Been Infringed and I Can't Sue!: New Approaches to Copyright Registration*, 41 U. MEM. L. REV. 449, 498 (2011).

not have to wait for the Copyright Office to make a determination regarding their work while the infringer continues to profit from the programmer's work, and, under both approaches, programmers have equal rights.[22] Based on the language and history of the Copyright Act, registration requires fewer formalities.[23] Under the application approach, Courts would become more efficient and predictable, providing programmers with the opportunity to enforce their legal rights.[24]

Part I of this Comment will provide background on application programs and describe the process to create an application program.[25] Part II will discuss what is copyright and then will begin with the history of the copyright law with respect to computer programs.[26] Then, this Comment will explain the elements of copyright protection and the limitation to the copyright scope for computer programs by discussing the primary test the Courts have adopted to differentiate the ideas from the expression of an idea of a work, specifically the copyright scope of the non-literal elements of computer programs.[27] Part III puts forth the elements required to prove copyright infringement.[28] Even though the plaintiff may be able to prove first element of copyright infringement, he or she may fail to prove the claim because the expressions have merged with the ideas.[29] Part IV of this Comment explains the circuit split over the registration requirements for litigation purposes.[30] Finally, this Comment discusses a proposed solution, which would provide programmers with greater copyright protections by repealing the merger doctrine and applying the application approach under the registration requirement for filing a suit.[31] As a consequence, programmers will have an expectation

---

22 2 MELVILLE B. NIMMER & DAVID NIMMER ON COPYRIGHT § 7.16[B][3][b][ii] (2017), Lexis (database updated Nov. 2017); *see also* Rita Marie Cain, *Timing Is Everything: Copyright Registration and Preregistration*, 88 J. PAT. & TRADEMARK OFF. SOC'Y 381, 386 (2006) (explaining how the application "reconciles other provisions in the federal statutory scheme.").

23 Erin Hogan, *Approval versus Application: How to Interpret the Registration Requirement under* the Copyright Act of 1976, 83 DENV. U. L. REV. 843, 860 (2006); Cosmetic Ideas, Inc. v. IAC/InteractiveCorp, 606 F.3d 612, 618 (9th Cir. 2010) (registering is optional based on the text and the history of the Copyright Act).

24 Jonathan L. Kennedy, *Double Standard and Facilitated Forum Shopping: A Historical Approach to Resolving the Circuit Split on Copyright Registration Timing*, 60 DRAKE L. REV. 305, 344–45 (2011); Mose Bracey, *Searching for Substance in the Midst of Formality: Copyright Registration as a Condition Precedent to the Exercise of Subject-Matter Jurisdiction by Federal Courts over Copyright Infringement Claims*, 14 J. INTELL. PROP. L. 111, 143 (2006).

25 *Infra* Part I.

26 *Infra* Part II.

27 *Infra* Part II.

28 *Infra* Part III.

29 *Infra* Part III.

30 *Infra* Part IV.

31 *Infra* Part IV.

of the type of protection for their programs.[32]

I. BACKGROUND ON APPLICATION PROGRAM

An average person's colloquial use "application program" is often interchangeable with "computer program."[33] In this Comment, "application program" and "computer program" have different meanings.[34] An "application program" is a program that allows "a user to perform some particular task such as word processing."[35] On the other hand, a "computer program" is a "set of instructions or steps which causes a computer to perform an underlying process, idea, or

---

32 *See* HOWARD B. ABRAMS, § 10:8.30 APPLICATION AND REGISTRATION APPROACHES TO SECTION 411(a), THE LAW OF COPYRIGHT (2017); Shaeffer, *supra* note 11, at 331 (explaining most of what copyright law protects should cover computer programs).

33 1–2 DAVID BENDER, COMPUTER LAW § 2.06, 2 (2017), Lexis (database updated Dec. 2017). *See generally* Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281, 343 (1970). A program is a set of instructions or statements for the computer to execute. ANDREW VAZSONYI, INTRODUCTION TO DATA PROCESSING 289 (Richard D. Irwin, Inc. 3d ed. 1980). A computer is a machine that "accepts instructions that tell the computer how to perform a desired function" and "take in and process the information." VERGARI & SHUE, *supra* note 12, at 13; WILLIAM S. DAVIS, INFORMATION PROCESSING SYSTEMS 48 (Addison-Wesley Pub. Co. 2d ed. 1981) (analogizing how a computer works with the manager's office between the two secretaries' offices with one receiving all of the information and communicating to the manager, the input of the computer, and the other one conveying the manager's information to the outside world, the output of the computer); Nicholas A. Holton, *Google, Inc. v. Oracle America, Inc.: Supreme Court Declines to Review Reversal of Landmark API Copyright Decision*, 62 LOY. L. REV. 189, 195 (2016) (describing how an application program works by the interaction of the software, which is the computer program, and the hardware, which is the computer).

34 *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 43 (D. Mass. 1990) (explaining how an application program is a type of computer program).

35 *Id.*; *see also* Menell, *supra* note 16, at 1051 (explaining how application programs "perform specific data processing tasks for the user" and examples of application program include statistical and financial analysis and video game programs).

algorithm."[36] The programmer can divide the computer program into the operating system program[37] and the application program.[38]

The purpose of an application program ("program") is to solve the particular problem.[39] A program is written by either a programmer is under an owner's direction to create the program, or an entity that leases or creates it for a price.[40]

A. The Process of Creating a Program

There are five steps in developing a program: (1) define the problem; (2) design the structure; (3) write the code, (4) debug and test; and (5) documentation and maintenance.[41]

---

36 VERGARI & SHUE, *supra* note 12, at 508; *see also* Velasco, *supra* note 5, at 244 (defining a computer program as a "list of instructions that the computer executes to achieve the result desired by the programmer."); *Computer Program*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984) (stating a computer program is a "sequence of instructions suitable for processing by a computer."); *cf.* 17 U.S.C. § 101(a) (2010) (defining "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.").

37 The operating system program or system software "are programs that control the basic functions of the computer hardware*." Lotus Dev. Corp.*, 740 F. Supp. at 43. An operating system "supervises and controls the over operation of a computer" and "gives the computer the intelligence to perform the work it is instructed by other programs to do." VERGARI & SHUE, *supra* note 12, at 20. The operating system, such as Windows "runs" the application program, Microsoft Word. Lothar Determann, *Dangerous Liaisons – Software Combinations as Derivative Works - Distribution, Installation, and Execution of Linked Programs Under Copyright Law, Commercial Licenses, and the GPL*, 21 BERKELEY TECH. L.J. 1421 (2006). The function of a system program or system software "is to make the computer work." An operating system, such as OS X High Sierra used on Apple Macintosh computers, is a system program. 1–2 BENDER, *supra* note 33, at § 2.06, 2–3. The discussion of this paper will focus on application programs.

38 Howard Root, *Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test*, 68 MINN. L. REV. 1264, 1268 (1984). An application program or software is "[a] computer software program designed for a specific job." *Application Program*, NEWTON'S TELECOMM DICTIONARY (28th ed. 2014). An application program can also be described as "task-specific software directed toward performance of a single function." VERGARI & SHUE, *supra* note 12, at 22. For example, an application program, such as Tetris, "performs specific data processing tasks for the [player]" by removing cubes when the same color cubes line up in a column.

39 1–2 BENDER, *supra* note 33 (stating examples of applications include word processors, video games, and media players). An application program is based on "what the user wants to do with the computer at any given time." Pamela Samuelson, *CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine–Readable Form*, 1984 DUKE L.J. 663, 686 (1984).

40 1-2 BENDER, *supra* note 33; *see also* YOHE, *supra* note 12, at 224 (explaining how communication between the programmer who writes the program and the originator who developed the idea of the program is essential to a successful program).

41 YOHE, *supra* note 12, at 222 (stating that programming is the process of "communication between the humans and the computer."). Computer programming is "the entire series of steps involved in solving a problem on a computer." SCHNEIDER ET AL., *supra* note 2, at 1.

1. Defining the Problem

The first step is to determine "what purpose the [program] will serve for the user."[42] The programmer has to analyze the problem by gathering and examining information required to produce the aspired result.[43] The programmer must think and rethink the problem and take into account the capabilities of the computer because he or she will clarify the problem and "make it take shape."[44] Once the programmer has decided what the program will do, he or she has to create written specifications, which is a "precise statement of the problem" that includes input and output data[45] specifications.[46]

2. Designing the Structure

Once the programmer decides the program's purpose, he or she defines the

---

Programming is the process of "designing, writing, and testing" the program. *Programming*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984). In creating a program, the programmer should be aware of the "target computer system" and the "programming environment." Wendy Seltzer, *Software Patents and/or Software Development*, 78 BROOKLYN L. REV. 929, 948 (2013).

42 Velasco, *supra* note 5, at 245; *see also* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993) (describing "the program's function as specifically as possible without reference to the technical aspects of the program"); VERGARI & SHUE, *supra* note 12, at 24; Duncan M. Davidson, *Protecting Consumer Software: A Comprehensive Analysis*, 1983 ARIZ. ST. L.J. 611, 619 (1983) (stating software development is first begun by defining the algorithm, which is defined as a step–by–step procedure used to solve problems).

43 HERBERT D. LEEDS & GERALD M. WEINBERG, COMPUTER PROGRAMMING FUNDAMENTALS 64 (McGraw-Hill, 1961). It may seem trivial to programmers to think about what they want to create, and this phase may seem "obvious." This step is "often overlooked or omitted by programmers." The programmer must have a "clear understanding of exactly what is needed is absolutely necessary for creating a workable solution." SCHNEIDER ET AL., *supra* note 2, at 2, 6.

44 SCHNEIDER ET AL., *supra* note 2, at 7; *see, e.g.*, E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1488 (D. Minn. 1985) (describing the development of "a trunked mobile radio system" by first creating the LTR code).

45 *See* 1–2 BENDER, *supra* note 33, at § 2.06 [3][b] (illustrating the input data refers to the format and the output data refers to specifications); *see also* Menell, *supra* note 16, at 1051 (stating programs perform specific data processing tasks for the user).

46 YOHE, *supra* note 12, at 221. The programmer also has to conduct a requirement analysis, which is a "careful formulation of the user's requirements so that the intent of the desired system, the properties it must possess, and the constraints on it are well understood by both the user and the system developer." Then, the programmer has to determine whether his or her program can meet the system specifications for the computer to run the program. PHILIP GILBERT, SOFTWARE DESIGN AND DEVELOPMENT 9 (Science Research Associates, Inc., 1983).

purpose in a series of steps.[47] The programmer will describe those steps by writing a pseudo code[48] or creating a flowchart.[49] To convey that representation, the programmer has to design the structure by creating data structures[50] and algorithms.[51] Data structures[52] are variables that hold data, and an algorithm takes the data structure, analyzes it, and has the computer execute it.[53] An algorithm is a step-by-step list of instructions to execute the purpose of the program, and the programmer represents an algorithm through a pseudo code or a flowchart.[54] The

---

47 VAZSONYI, *supra* note 33, at 291; *see also* YOHE, *supra* note 12, at 225 (explaining how "[o]nce the problem has been identified, the next step is to determine the methods to be used in solving it.").

48 A pseudo code is "a representation of program structure consisting of written statements that resemble actual source code." 4 MELVILLE B. NIMMER & DAVID NIMMER ON COPYRIGHT § 13.03[F][1] n.290 (2017); *see also* Siebersma v. Vande Berg, 64 F.3d 448, 449 (8th Cir. 1995) (explaining how the defendant translated the handwritten programming commands or pseudo code into a computer language); LAWRENCE J. PETERS, SOFTWARE DESIGN: METHODS & TECHNIQUES 93 (Yourdon Press, 1981) (describing a visual of pseudo code, which does not consist of syntax or semantics).

49 Swinson, *supra* note 10, at 148. A flowchart is a "graphical representation of the definition, analysis, or solution of the problem in which symbols are used to represent such things as operations, data flow, and equipment." *Flowchart*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984). The flowchart is the foundation for writing the program. Martyniuk, *supra* note 11, at 1342; *see also* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 43–45 (D. Mass. 1990) (explaining how a flowchart can be a literal expression of the program and is "a symbolic outline or scheme outline" of the program's logic); VAZSONYI, *supra* note 33, at 151 (providing an illustration of a flowchart).

50 Data structures are "an organization of data" that allow an algorithm to work and how "[o]ne part of a program organizes the data in such a way that the other part, which embodies the algorithm, can work with it." Marci A. Hamilton & Ted Sabety, *Computer Science Concepts in Copyright Cases: The Path to a Coherent Law*, 10 HARV. J.L. & TECH. 239, 252, 254 (1997).

51 SCHNEIDER ET AL., *supra* note 2, at 3, 19. An algorithm is "a series of steps telling a processor how to perform a task." Swinson, *supra* note 10, at 146. In the design stage, "the programmer [creates] a series logical steps, depicted in a flow chart, for the solution of the problem – this series of steps constitute the program's algorithm." VERGARI & SHUE, *supra* note 12, at 24.

52 Data structures are "a precise representation or specification of a data type." Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993). For instance, if a program generates a list of employees who have punched in and out, one of the variables would be employee name. The purpose of data structures is to "reduce[] the complexity of the program." YOHE, *supra* note 12, at 226; *see also* Hamilton & Sabety, *supra* note 50, at 255 (presenting an example of a data structure).

53 Shaeffer, *supra* note 11, at 349. For example, Expedia.com, a travel booking website, offers information on hotel and flight prices, and the algorithm collects and organizes the user's data from sellers and generates a relevant list of sellers based on the consumer's requirements. *See* Michael S. Gal & Niva Elkin-Koren, *Algorithmic Consumers*, 30 HARV. J.L. & TECH. 309, 313–14 (2017).

54 YOHE, *supra* note 12, at 228.

An algorithm can be viewed "as a recipe that tells [the programmer] exactly how to go about getting desired results." A recipe on how to make a chocolate cake is an example of an algorithm. SCHNEIDER ET AL., *supra* note 2, at 19. When the programmer writes the algorithm in

programmer has to decompose the algorithm into subroutines.[55] Subroutines are a list of instructions that perform one aspect of the program, and break down the subroutines into subtasks to create its own algorithm and subroutines continue to do this process until it is feasible for the programmer to implement.[56] The programmer produces a flowchart or a pseudo code to assist the programmer with designing and visualizing the program.[57]

3. Writing the Code

The programmer translates the problem in the form of a pseudo code or a flowchart into a machine-readable language to "direct the computer through the process of solving the problem."[58] This process is known as coding.[59]

First, the programmer implements[60] the pseudo code or flowchart by writing each step of the algorithm in a programming language.[61] This is because the

---

pseudo code or a flowchart, he or she describes the program at a high level "with no particular processor in mind." Swinson, *supra* note 10, at 148.

55 Susan A. Dunn, *Defining the Scope of Copyright Protection for Computer Software*, 38 STAN. L. REV. 497, 500 (1986) (stating a subroutine is "a set of computer instructions that performs a specific computational procedure."). A subroutine is "sequence of instructions for a specific function that is often called by a program." *Subroutine*, NEWTON'S TELECOMM DICTIONARY (Steve Schoen 28th ed. 2014); Arenault, *supra* note 9, at 147 (illustrating an example of a subroutine). A module is "a collection of logically related code that is part of a program." Module consists of many subroutines, and when executed, the process of input to output occurs. JOAN K. HUGHES & JAY I. MICHTOM, A STRUCTURED APPROACH TO PROGRAMMING 21, 23 (Prentice-Hall, Inc. 1977).

56 LEEDS & WEINBERG, *supra* note 43, at 212; Velasco, *supra* note 5, at 246.

57 VAZSONYI, *supra* note 33, at 292; SCHNEIDER ET AL., *supra* note 2, at 34.

58 VERGARI & SHUE, *supra* note 12, at 24–25; Breyer, *supra* note 33, at 341–42 (describing the translation of the algorithm, which is embodied in a flowchart, into a series of instructions in a programming language); *see also* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993); Data Cash Sys, Inc. v. JS&A Group, Inc. 480 F. Supp. 1063, 1065 (N.D. Ill. 1979) (explaining the steps of programming including "development of a 'source program' which is a translation of the flowchart into computer programming language.").

59 YOHE, *supra* note 12, at 231; *see also* VAZSONYI, *supra* note 33, at 292 (defining coding as "the process of translating program flowcharts into source programs."). Coding can also be known as implementation because the programmer implements the program's design by coding. Velasco, *supra* note 5, at 246.

60 The programmer takes the design of the program in flowchart form and implements it into machine-readable code. Menell, *supra* note 16, at 1055; *see, e.g.*, Richard H. Stern, *Copyright in Computer Programming Languages*, 17 RUTGERS COMPUTER & TECH. L.J. 321, 370 (1991) (explaining how the programmer who is "skilled and persistent" may find it "easier to implement a list processing procedure in LISP than FORTAN.").

61 A programming language is "a set of symbols and syntax rules used to express strings of bits." Holton, *supra* note 33, at 196. A programming language is "a tool for writing computer programs." Dennis S. Karjala, *Oracle v. Google and The Scope of a Computer Program Copyright*, 24 J. INTELL. PROP. L. 1, 16 (2016). A programming language can be defined as a

computer cannot execute[62] a human language since it does not understand it.[63] The type of programming language a programmer chooses reveals the level of training as well as the style and creativity.[64] A programmer can write the program in different programming languages, such as machine,[65] assembly,[66] and high-level.[67]

program's written instructions. "Each programming language has a unique grammar and set of meanings." John W.L. Ogilvie, *Defining Computer Program Parts under Learned Hand's Abstractions Test in Software Copyright Infringement Cases*, 91 MICH. L. REV. 526, 531 (1992).

62 Execution is "to change the state of a computer in accordance with the rules of the operations it recognizes." AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Aufrichtig, *supra* note 15, at 146 (explaining how "all programs that are executed by computers must be in machine language before the computer can understand them."); Steven R. Englund, *Idea, Process, or Protected Expression: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 868 n.12 (1990) (stating "an instruction is 'executed' when a computer performs that instruction.").

63 Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230 (3d Cir. 1986); VAZSONYI, *supra* note 33, at 103; *see also* Dunn, *supra* note 55, at 501 (explaining how computers operate on a machine language and not on a human language).

64 Anthony L. Clapes, Patrick Lynch & Mark R. Steinberg, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. REV. 1493, 1535 (1987); ERIK PIÑEIRO, THE AESTHETICS OF CODE: ON EXCELLENCE IN INSTRUMENTAL ACTION 25 (2003); *see also* Pamela Samuelson, Randall Davis, Mitchell D. Kapor & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2316 (1994) (stating how programs have not only textual material but also behavior characteristics, such as copying and pasting text).

65 A machine language consists of instructions in zeros and ones. *Copyright Protection of Computer Program Object Code*, 96 HARV. L. REV. 1723, 1724 (1983); *see also* Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (explaining how machine language is in the form of a binary language using symbols – zeros and ones, which indicate an off and on switch. For example, 01101001 means adding two numbers and saving the result).

66 An assembly language is "written in simple symbolic names, or alphanumeric symbols, [which is] more easily understandable by human programmers." For example, "LOAD B; DIV C; ADD A" means take B and divide it by C and add A. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 44 (D. Mass. 1990).

67 Hamilton & Sabety, *supra* note 50, at 265–66. A high-level language is a "programming language that does not reflect the structure of any one given computer or that of any given class of computers." *High-level language*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see, e.g.*, Root, *supra* note 38, at 1266 n.12 & 13 (explaining how programmers use high–level languages, such as FORTRAN (Formula Translation) for mathematical statements or COBAL (Common Business Oriented Language) for business data processing).

However, most programmers code[68] in a language that is readable by humans.[69] Then, he or she uses a compiler[70] or an interpreter,[71] depending on the programmer's needs,[72] to convert the program in a source code[73] into an object code,[74] which computers can read.[75] Once the programmer has produced an object code, the computer can execute the written instructions and produce the desired result.[76]

i. Machine

A machine language is "a programming language composed of machine in-

68 A code is "a set of unambiguous rules specifying the manner in which data may be represented in a discrete form." *Code*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984).

69 SCHNEIDER ET AL., *supra* note 2, at 202; *see also* Velasco, *supra* note 5, at 244 (explaining how it is easier to write in a programming language that understandable); Englund, *supra* note 62, at 868 (stating how "computers are only capable of directly executing 'computer instructions'").

70 A compiler "translate[s] a computer program expressed in a problem-oriented language into a computer-oriented language." *Compile*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* VERGARI & SHUE, *supra* note 12, at 26 (defining a compiler as translating a high-level language into a machine code).

71 An interpreter translates one statement at a time from source code to object code. Mislow, *supra* note 12, at 744 n.45.

72 While a compiler translates the "entire text of a high-level program in one continuous process" and then runs the program, an interpreter translates "one statement at a time" and then runs each statement. Mislow, *supra* note 12, at 744 n.45; *see also* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 44 (D. Mass. 1990) (describing that choosing between an interpreter or a compiler depends on the type of programming language and CPU). A CPU, or computer processing unit, is the "essential core of the computer system" and are the computer's "brains." VERGARI & SHUE, *supra* note 12, at 7; Vaidehi Joshi, *A Deeper Inspection into Compilation and Interpretation*, MEDIUM (Dec. 13, 2017), https://medium.com/basecs/a-deeper-inspection-into-compilation-and-interpretation-d98952ebc842 (explaining the differences between a compiler and an interpreter).

73 A source code is defined as statements written in a high-level language. Apple Comput., Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983); *see also* Christina M. Reger, *Let's Swap Copyright for Code: The Computer Software Disclosure Dichotomy*, 24 LOY. L.A. ENT. L. REV. 215, 219 (2004) (describing source code to include "descriptive words, formulas, and mathematical equations."). *See generally* SCHNEIDER ET AL., *supra* note 2, at 202 (writing in high-level languages as a matter of convenience and familiarity, requiring translation from source code into object code).

74 An object code is an "[o]utput from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code." *Object Code*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Dunn, *supra* note 55, at 501 (explaining how an object code as "either [an] executable machine code or code that can be processed easily into machine code.").

75 VAZSONYI, *supra* note 33, at 103.

76 Ogilvie, *supra* note 61, at 531; Samuelson, *supra* note 39, at 686.

structions that can be executed directly by a computer without further compilation" and is the only language recognized by a machine.[77] It represents the instructions in binary form, consisting of zeros and ones.[78] For example, the instruction 010001 could mean run program.[79]

Programmers usually do not write their algorithms in machine language because it is difficult for them to understand and it is tedious.[80] Instead, a programmer writes the program in a readable language and produces a source program.[81] Then, he or she converts the source program into an object program,[82] which consists of strings of binary digits.[83] Upon conversion, the computer reads the language, executes the algorithm, and produces the expected result.[84]

ii. Assembly

An alternative to machine language is the assembly language for writing his

---

77 *Machine language*, NEWTON'S TELECOMM DICTIONARY (28th ed. 2014); Aufrichtig, *supra* note 15, at 146 (explaining how "difficult [it is] too become familiar with and each instruction does only a small task because it takes seven machine language instructions to execute C= A + B.").

78 Clapes et al., *supra* note 64, at 1521; *see also* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 43 (D. Mass. 1990) (describing the expression of machine instructions in a hexadecimal form consisting of numbers and letters).

79 Mislow, *supra* note 12, at 743; *see, e.g.*, Davidson, *supra* note 42 (representing "100110010" as an instruction to add "register B to whatever is in the accumulator.").

80 *Copyright Protection of Computer Program Object Code*, *supra* note 65, at 1725; *see also Lotus*, 740 F. Supp. at 44 (explaining how programmers rarely write directly on the computer); YOHE, *supra* note 12, at 236 (describing that choosing a programming language is based on one's experience and natural familiarity).

81 VERGARI & SHUE, *supra* note 12, at 25. A source program is a "computer program that sorts items of data." *Source Program*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Data Cash Sys, Inc. v. JS&A Group, Inc. 48 F. Supp. 1063, 1065 (N.D. Ill. 1980) (stating computer scientists use source program and source code interchangeably).

82 An object program is same as an object code, which is "a program written in machine language that can be directly executed by the computer's CPU without need for translation," because both the program and the code are a set of instructions that the computer executes. *Lotus*, 740 F. Supp. at 43; *see also Object Code*, NEWTON'S TELECOMM DICTIONARY (28th ed. 2014) (defining an object code can also be defined as an "instruction code in machine language.").

83 Reger, *supra* note 73, at 219; Root, *supra* note 38, at 1267. A binary code consists zeros and ones, and a binary digit or a bit is in the form of a zero or a one. *Binary code & binary digit*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984). "A set of bits" is called a byte. *Byte*, NEWTON'S TELECOMM DICTIONARY (28th ed. 2014). When the programmer produces an object code or a binary code, the machine reads the strings of bits or lines of binary code to produce the end result. Holton, *supra* note 33, at 196.

84 Samuelson, *supra* note 39, at 686; *see also* VAZSONYI, *supra* note 33, at 103, 329 (explaining how, after writing the code, the instructions are "inputs to the computer" and, after the translation of the source program, the programmer has to produce a "machine language object program," which the computer executes and produces the desired resulted).

or her program.[85] Assembly language is represented in mnemonic or alphanumeric instructions.[86] For example, the instruction "ADD X" could mean Add lane X to the register.[87] In order for the computer to execute the program or the series of algorithms, the programmer has to translate the assembly language into machine code using an assembler.[88]

iii. High Order

The most popular and widely used programming language is the high-level language because it uses "English-like"[89] words and is easier to understand and program[90] than the machine or assembly language.[91] High-level language is "a programming language that does not reflect the structure of any one given computer or that of any given class of computers."[92] The programmer translates the algorithms that are components of the program in a high-level language into a

85 Shaeffer, *supra* note 11, at 351 (stating each "assembly language instruction corresponds to a single machine-language instruction."); Dunn, *supra* note 55 (explaining how "[humans] can read assembly language" but it resembles closer machine language).

86 *Copyright Protection of Computer Program Object Code*, *supra* note 65, at 1725; Aufrichtig, *supra* note 15, at 147. Another closely related programming language is symbolic language, which "expresses addresses and operation codes of instructions in symbols." *Symbolic language*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984).

87 Davidson, *supra* note 42, at 620.

88 Root, *supra* note 38, at 1267; VAZSONYI, *supra* note 33, at 104 (explaining an assembler is a program that "translates and assembles the assembler language program into a machine language program."); *see also* Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 844 n.6 (Fed. Cir. 1992) (explaining how a disassembly program translates a machine language into assembly language and is the reverse of an assembler).

89 Davidson, *supra* note 42, at 620; Velasco, *supra* note 5, at 244; *see* SCHNEIDER ET AL., *supra* note 2, at 205 (illustrating high-level language in easy and understandable words).

90 Program means "[t]o design, write, and test computer programs." *Program*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Ogilvie, *supra* note 61 (defining a program, or a software that runs on the computer, as "an organized set of instructions that guides a computer.").

91 Dunn, *supra* note 55 (explaining how high–level languages are "designed to accommodate the programmer"); *see also* Bradley Nice, *A Complete List of Computer Programming Languages*, MEDIUM (Mar. 18, 2017), https://medium.com/web-development-zone/a-complete-list-of-computer-programming-languages-1d8bc5a891f (explaining various high-level languages, such as Algol, BASIC, Pascal, C, C++, Java, Python, Perl, and Ruby).

92 *High-level language*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also Copyright Protection of Computer Program Object Code*, *supra* note 65, at 1725 (explaining "[e]ach high-level language instruction is the equivalent of several assembly or machine language instruction."); Menell, *supra* note 16, at 1056 (stating while there are various types of high-level languages, what matters is the "accuracy, efficiency, and reliability of the resulting program.").

machine code because the computer "can only deal with machine language programs written in bit strings."[93] Once the translator[94] converts the source program into an object program, the program is machine readable.[95] Then, the programmer can run the program.[96]

4. Debugging and Testing

Before the programmer begins to debug[97] and test[98] the program, he or she should review and update the problem statement, algorithms, data structures, and pseudo code or flowchart to reaffirm his or her problem produces the intended result.[99]

Once the programmer reviews and proofreads the program, he or she runs the

---

93 Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230-31 (3d Cir. 1986); VAZSONYI, *supra* note 33, at 103; Swinson, *supra* note 10, at 149.

94 A translator is a "computer program that translates from one language into another language and in particular from one programming language into another programming language." *Translator*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 44 (D. Mass. 1990) (explaining how a programmer can translate using an interpreter or a complier).

95 Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 838 (Fed. Cir. 1992); Dunn, *supra* note 55, at 501.

96 Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform*, 7 UCLA J.L. & TECH. 1, 6–7 (2005). To run a program means a "performance of one or more programs." *Run*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* VAZSONYI, *supra* note 33, at 293 (stating how after producing an object program and running the program, the programmer will see whether the program produces the result).

97 Debug means "to detect, to trace, and to eliminate mistakes in computer programs or in other software." *Debug*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Menell, *supra* note 16 (debugging is a process that involves "testing the program for accuracy, correcting programming errors, and verifying that the program functions properly.").

98 Testing means comparing the produced results to the expected results. GILBERT, *supra* note 46, at 13; *see also* HUGHES & MICHTOM, *supra* note 55, at 237 (explaining how testing is a positive activity while debugging is a negative one).

99 YOHE, *supra* note 12, at 236, 239–40.

program to determine the errors[100] or bugs[101] in the program.[102] Debugging consists of find and correcting all errors causing the program to produce "either incorrect results or no results."[103] Testing is a process "in which a program is validated" and "the how and what to test can be specified."[104] When the programmer is debugging and testing, he or she is concerned with finding and correcting all possible errors.[105] While the programmer is searching for and correcting bugs, he or she should record all bugs because he or she can determine other places bugs may be found.[106] The programmer continues this process until all known

100 An error is "[a] discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition." *Error*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Velasco, *supra* note 5, at 246 (stating that finding and correcting the errors in the program does not end when the program is distributed because the programmer has to continue to maintain the program by correcting "any hidden errors and improv[ing] any imperfections.").

101 A bug is a "problem in a software." *Bug*, NEWTON'S TELECOMM DICTIONARY (28th ed. 2014). Bugs have the same meaning as errors because they are errors in a program. LEEDS & WEINBERG, *supra* note 43, at 66.

102 YOHE, *supra* note 12, at 236 (stating the programmer can conduct the compilation process before testing). *See generally* MARC GREGOIRE, PROFESSIONAL C++ 829 (Robert Elliott et al. eds., 3d ed. 2014) (describing in depth procedures for isolating and correcting bugs as well as guidance on avoiding the bugs altogether).

103 SCHNEIDER ET AL., *supra* note 2, at 204. The debugging procedure has two phases, which are to "get the program running," regardless of whether the program "outputs valid data," and to ensure the program runs properly. First, the programmer is concerned with figuring out the "bugs" of the program. In the second phase, it is difficult to discern whether the program will run as expected because the programmer has to compare the flow chart to the listing and has to determine the program will run the instructions the programmer provided in the flowchart. 1–2 BENDER, *supra* note 33, at § 2.06 [3][f]. *See generally* GREGOIRE, *supra* note 102, at 827–61 (describing the use of flow charts in the debugging process as well as a comprehensive overview of debugging in C++).

104 HUGHES & MICHTOM, *supra* note 55, at 237 (stating that debugging cannot begin until the programmer has an executable program, while testing can begin earlier); *see also* Alex Bachuk, *Understanding software testing*, MEDIUM (Oct. 6, 2017), https://medium.com/@netxm/how-to-get-started-with-software-testing-9fa1ce4f2a64 ("[T]esting as a user and as a developer are white box and black box texting" with the former "inspect[ing] source code and verify[ing] it works according to the spec" and the latter access "only text [from the] external interfaced produced by the code.").

105 Samuelson, *supra* note 39, at 687; *see also* VAZSONYI, *supra* note 33, at 293 (explaining how debugging and testing is not only about finding errors but also achieving the intended result); GREGOIRE, *supra* note 102, at 842–43 (detailing additional steps to reproduce bugs using the smallest possible sequence and showing that narrowing the sequence first allows the bug to be efficiently isolated and corrected).

106 HUGHES & MICHTOM, *supra* note 55, at 247; YOHE, *supra* note 12, at 238.

errors are removed.[107] This stage is probably the most frustrating and time-consuming task to the programmer.[108]

If the programmer believes he or she has removed all known bugs, then he or she can run the program to see whether the computer produces the desired result.[109] When the program fails to run as expected, the programmer has to restart the debugging and testing process.[110] After each run, the programmer should thoroughly examine the results and determine what corrections must be made.[111] If the programmer believes the program is running properly and is producing the expected result, then he or she can verify the program.[112] The programmer continues this process of finding all conceivable errors even after dissemination of the program.[113]

5. Documenting and Maintaining

Even though the programmer may have removed the necessary errors, he or she cannot distribute the program to the public because he or she has to document or explain the program.[114] During documentation, the programmer is "creating, collecting, organizing, storing, and communicating information necessary

---

107 1–2 BENDER, *supra* note 33, at § 2.06 [3][f]; *see* VERGARI & SHUE, *supra* note 12, at 1 (explaining how computers only produces the programmer's desired result if the program is error or bug free; otherwise, the computer will stop or malfunction).

108 SCHNEIDER ET AL., *supra* note 2, at 204 (explaining that with large programming systems, studies show that fifty to seventy-five percent of the time of programming is spent on debugging). When a programmer encounters a defect that is difficult to remove, he or she will examine "the operation of an object code program in minute detail using a run-time debugger to set break points and single step through object code instructions." Michael F. Morgan, *The Cathedral and the Bizarre: An Examination of the Viral Aspects of the GPL*, 27 J. MARSHALL J. COMPUTER & INFO. L. 349, 413 (2010).

109 HUGHES & MICHTOM, *supra* note 55, at 237; *see* John M. Conley & Robert M. Bryan, *A Unifying Theory for the Litigation of Computer Software Copyright Cases*, 63 N.C.L. REV. 563, 566 (1985) (editing the program continues even after distribution).

110 HUGHES & MICHTOM, *supra* note 55, at 237; *see also* GREGOIRE, *supra* note 102, at 842 (describing multiple steps can be required when attempting to reproduce and eliminate a coding error during the debugging process).

111 YOHE, *supra* note 12, at 238; *see* Quad County Distrib. Co. v. Burroughs Corp., 385 N.E.2d 1108, 1100–11 (D. Ill.1979) (stating the programmer has to initiate a two-step debugging process to eliminate the defects in the program by "applying hypothetical data to the program; when it appears the defect have been removed" and "actual data is used to assure all defects have, in fact, been removed.").

112 YOHE, *supra* note 12, at 239 (stating that verifying can be done by running the program on a different computer or an independent calculation).

113 HUGHES & MICHTOM, *supra* note 55, at 237; 1–2 BENDER, *supra* note 33, at § 2.06 [3][f] (explaining how coding changes are required to remove the bugs and must be careful to not introduce new bugs).

114 SCHNEIDER ET AL., *supra* note 2, at 5. The programmer has already created most of the documentation during the problem definition, the design of the structure, writing the code, and debugging and testing steps. YOHE, *supra* note 12, at 240.

to use the program."[115] The purpose of documentation is to communicate the program to other programmers who are interested in modifying the program.[116] Programmers should also make comments and add descriptions to convey "an understanding of the program," because they may forget what they did to the program when they return to it to correct any errors or make any improvements.[117]

While the program is available to the public, the programmer has to continue to maintain the program by noting the existence of any bugs and correcting them.[118] Having good documentation will assist the programmer with maintaining the program because he or she can use it to understand why a specific part of the program is not functioning as expected or to make any new improvements to the program.[119] After the programmer makes the necessary changes, he or she should update the documentation to reflect any bug-fixes and enhancements per new features.[120]

Once the program is functional, the user does not see the code or the design

---

115 VAZSONYI, *supra* note 33, at 293–94. Documentation is defined as "[t]he aids provided for understanding the structure and intended uses for an information system or its components, such as flowcharts, textual material, and end-user manuals" and the "management of documents" required for "identifying, acquiring, processing, storing, and disseminating them" for the program. *Documentation*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984). The programmer creates two types of documentations: the technical documentation is "a description of the technical details" and other programmers who modify the program require this documentation and the user documentation provides "information that one needs in order to use [the] program" and explains to the user how to use the program. SCHNEIDER ET AL., *supra* note 2, at 226–27.

116 1–2 BENDER, *supra* note 33, at § 2.06 [3][g]; *see also* Menell, *supra* note 16 (documenting entails "preparing materials that explain the functioning of the program.").

117 YOHE, *supra* note 12, at 222, 233; *see also* Lionel M. Lavenue, *Technical Data Rights in Government Procurement: Intellectual Property Rights in Computer Software and the Indicia of Information Systems and Information Technology*, 32 U.S.F. L. REV. 1, 48 (1997) (explaining how "documentation includes owner's and user's manuals, installation and operating instructions, and other descriptive materials for computer software.").

118 YOHE, *supra* note 12, at 240–41; *see also* SCHNEIDER ET AL., *supra* note 2, at 5 (explaining how programs are not "static entities" and become "outdated" or "new equipment becomes available"). Maintenance is an activity where the programmer intends "to retain" the program as "functional" or "to restore it to, a state in which it can perform its required function." *Maintenance*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984).

119 YOHE, *supra* note 12, at 240–41; *see also* PIÑEIRO, *supra* note 64, at 127 (explaining how having a readable code makes it easier for the programmer to maintain).

120 SCHNEIDER ET AL., *supra* note 2, at 5.

and is concerned with utility and functionality of the program.[121] Many commentators and courts[122] view programming as an activity that involves achieving a particular result, thereby ignoring the process of creating a program.[123] From the programmer's perspective, courts fail to consider the programmer's code and design reveals much about the programmer, such as his or her skill and technical preference.[124] Essentially, the creation of a program is not "a scientifically objective process."[125] Instead, courts must understand the programmer put considerable time and creativity into creating the program by developing flowcharts or pseudo codes and designing algorithms.[126] As a consequence, courts cannot simply separate the design and the code of the program from the desired result because, to produce the idea, the programmer must write algorithms and flowcharts.[127]

---

121 David G. Luettgen, *Functional Usefulness vs. Communicative Usefulness: Thin Copyright Protection for the Nonliteral Elements of Computer Programs*, 4 TEX. INTELL. PROP. L.J. 233, 249 (1996); Samuelson, *supra* note 39, at 682; *see, e.g.*, Ilog, Inc. v. Bell Logic, LLC, 181 F. Supp. 2d 3, 10 (D. Mass. 2002) (stating the rule editors in the business rule computer software were an idea and thus were unprotected).

122 *See* Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1248 (3d Cir. 1986) (holding copyright protection may extend beyond the literal code and encompass the "structure, sequence, and organization"). *But see* Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 697 (2d Cir. 1992) (applying a substantially similar test and finding software derived from a previous job, but lacking identical elements of source code, did not infringe the original designer's copyright); Englund, *supra* note 62, at 881 (illustrating how courts can readily apply the Baker v. Selden rule by only looking at the work's purpose).

123 Reger, *supra* note 73, at 218–219; Shaeffer, *supra* note 11, at 342; *see also* Jonathan Ambrose, *Oracle American Inc. v. Google, Inc.: The Only Nonliteral Aspect of Java API's Protected Under Copyright Law are the Ones Nobody Wants to Copy*, 14 N.C. J.L. & TECH. ON. 1, 22 (2012) (stating that besides the functionality of a computer program, there is nothing left).

124 VERGARI & SHUE, *supra* note 12, at 509; Arenault, *supra* note 9, at 161; *see, e.g.*, Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 840 (Fed. Cir. 1992) (stating Nintendo's 10NES program was protected because it had "creative organization and sequencing unnecessary to the lock and key function and arranged the arbitrary programming instructions "in a unique sequence to create a purely arbitrary data system.").

125 Shaeffer, *supra* note 11, at 346; *see also* PIÑEIRO, *supra* note 64, at 242–43 (explaining how software is not an objective piece of mathematical code buy a subjective design that accomplishes a goal by one of many possible methods); Clapes et al., *supra* note 64, at 1499 (failing to understand "computer programs as a form of expression" by law and policymakers who view programs as having "inferior status as law" and deserve less protection).

126 Mislow, *supra* note 12, at 800; Lisa C. Green, *Copyright Protection and Computer Programs: Identifying Creative Expression in a Computer Program's Nonliteral Elements*, 3 FORDHAM INTELL. PROP., MED. & ENTM'T. L.J. 89, 135 (1992); *see also* Menell, *supra* note 16, at 1053–54 (explaining how the human factor assists in developing and expanding the design and the coding of application programs).

127 Matthew J. Faust, *What Do We Do with a Doctrine like Merger? A Look at the Imminent Collision of the DMCA and Idea/Expression Dichotomy*, 12 MARQ. INTELL. PROP. L. REV. 131, 150–51 (2008). When the court can only express the idea of the program in one or few ways, it will determine the program's idea and expression, such as source code and algorithm, have merged. Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir.

After the programmer is confident about his or her application program, the programmer can choose to disseminate it to the public.[128] When contemplating this decision, the programmer may have concerns about the unlawful use of his or her work.[129] Therefore, the programmer should obtain copyright protection by registering his or her work.[130] When the programmer files his or her program with the Copyright Office, he or she should ensure his or her work meets the requirements of copyright protections.[131]

II. WHAT IS COPYRIGHT?

Article I vests Congress with the power "[to] promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Investors the exclusive Right to their respective Writings and Discoveries."[132] While the United States Government has regulatory power over copyright protection and implemented the first Copyright Act in 1790, in 1909 Congress codified the common law and statutory copyright by the states into a single federal statutory system.[133] In 1988, Congress amended the Copyright Act of 1976 and added computer programs as a literary work.[134] Notably, copyright is not defined in Section 101 of

---

1983); *see* Dan L. Burk, *Method and Madness in Copyright Law*, 2007 UTAH L. REV. 587, 613 (2007) (stating that merging the software's processes, methods, and systems is incorrect because "they are quite literally geared to a particularized output."); *see, e.g.*, Apple Comput., Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1033 (N.D. Cal. 1983) (holding the D2 Macintosh design animation is unprotected because the idea of moving "a window to a new position" mergers with the expression of crafting the manipulation are inseparable).

128 YOHE, *supra* note 12, at 240–41; *see* Luettgen, *supra* note 121 (stating how programs exist to be functional, such as WordPerfect function for print, which prints pages when directed).

129 Velasco, *supra* note 5, at 248 (explaining how the user interface may not be protected).

130 17 U.S.C. § 411(a) (2010); *Copyright Basics*, U.S. COPYRIGHT OFF. (Sept. 2017), https://www.copyright.gov/circs/circ01.pdf; *see, e.g.*, M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 425, 429 (4th Cir. 1986) (stating the plaintiff registered its video game because it believed the defendant copied the audiovisuals and the machinations of the video characters).

131 17 U.S.C. § 102(a) (2010) (listing what Copyright protection includes and does not cover); 17 U.S.C. § 410(a) (2010); *Copyright Basics*, *supra* note 130.

132 U.S. CONST., art. 1, § 8, cl. 8.

133 28 U.S.C. § 1338(a) (2011) (stating the federal courts have "original jurisdiction" over copyright cases); 17 U.S.C. §§ 101–1332 (2010); H. Tomás Gómez-Arostegui, *Copyright at Common Law in 1774*, 47 CONN. L. REV. 1, 3 (2014); U.S. Copyright Office, *supra* note 3, at § 101.1; *see also* H.R. REP. NO. 60-2222, at 2, 7 (1907) (stating the Constitution did not create copyrights; instead, Congress has "the power to grant" the right to establish copyright. The first copyright statute passed was in Connecticut in 1783, but it became difficult for an author to obtain copyrightability because each state had different forms of protection).

134 U.S. COPYRIGHT OFFICE, *supra* note 3, at § 101.1; Public Law 95–517, Stat. 94 3015, 3028–29 (Dec. 12, 1980) (amending 17 U.S.C. § 102(a) to include computer program as eligible for copyright registration and protection); *see* S. REP. NO. 94-473, at 50 (1975) (explaining how Congress intended to protect computer programs based on the history of the copyright

the Copyright Act.[135]

A. History of the Copyright Act – Computer Programs

In 1974, Congress created the Commission on New Technologies Uses of Copyrighted Works (CONTU), which studied and made recommendations to Congress, regarding the copyright protection of computer programs.[136] After considering CONTU's recommendations, Congress amended the Copyright Act in 1980 and considered computer programs as a literary work, providing the same protections and limitations as other literary works.[137] The Copyright Act of 1976 defines a "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[138] While programmers may have copyright protection for their programs, the statute provides limited protection.[139]

---

law); NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1978) (recommending how to provide computer programs with copyright protection and to put forth its copyright scope). While this comment notes a difference between computer programs and application programs, the Copyright Act of 1976 treats them as the same; therefore, this comment for purposes of the copyright law will use the word "computer program."

135 17 U.S.C. § 101 (2010).

136 Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1247 (3d Cir. 1983); NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, *supra* note 134; *see also* Daniel J. Smith, *Proof of Copyright Infringement by Unauthorized Use of Software*, 52 AM. JUR. PROOF OF FACTS 3D § 14 (describing how Congress had incorporated the recommendations of CONTU into the 1980 Computer Software Copyright Act); Deborah F. Buckman, 180 A.L.R. FED. 1 *Copyright Protection of Computer Programs* § 2[a], Westlaw (database updated Jan. 2017).

137 Pub. L. No. 96-517, § 10, 94 Stat. 3015, 3028 (1980) (codified at 17 U.S.C. §101 (2010)); H.R. REP. NO. 96-1307, at 23 (1980); NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, *supra* note 134, at 63; *Apple Comput*., 714 F.2d at 1247 (explaining how "[l]iterary works are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied."); *see also* 96 CONG. REC. S30336 (daily ed. Nov. 20, 1980) (statement of Hon. Levin) (providing the purpose of the 1976 Copyright Act was to include copyright protection to computer software). A computer program is not patentable because the procedure to create a computer program is an idea, and Congress must decide whether to consider computer programs as patentable. Gottschalk v. Benson, 409 U.S. 63, 72, 73 (1972); VERGARI & SHUE, *supra* note 12, at 523 (explaining how "patent law can provide protection for the whole process" while copyright law provides protection to the image).

138 17 U.S.C. § 101 (2010).

139 Martyniuk, *supra* note 11, at 1340; *see* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 833 (10th Cir. 1993) (stating "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."). *See generally* 17 U.S.C. § 102(a) (2010) (providing computer software is considered copyrightable material).

B. The Requirements of Copyright Protection

For a program to have copyright protection, it must meet the elements of originality[140] and fixation.[141] However, the plain language of the statute does not state those requirements.[142] Nonetheless, courts have determined originality and fixation as constitutional and statutory requirements for copyright protection.[143] The purpose of copyright law was to protect the "original expression of any creative effort and not the effort itself."[144]

1. Originality

For a work to have originality, "the work owes its origin to the author."[145] Originality requires the creator to "independently create" and to "possesses at least some minimal degree of creativity."[146] While the subsequent work may be

---

140 Originality means "an original work of authorship in the expression of ideas." VERGARI & SHUE, *supra* note 12, at 552; *see, e.g.*, Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522, 523 (6th Cir. 2004) (establishing the requirement of originality because the Toner Loading Program met the "extremely low" standard).

141 U.S. CONST., art. 1, § 8, cl. 8 (stating a work's copyright requirements are based on the terms "Author" and "Writing" in the constitution); Harper & Row, Publrs., Inc. v. Nation Enters., 471 U.S. 539, 555 (1985); Dunn, *supra* note 55, at 507. Fixation means "a work that is fixed in a tangible medium from which it can be perceived for more than a short period of time." VERGARI & SHUE, *supra* note 12, at 552; *see, e.g.*, Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855 (2d Cir. 1982) (meeting the fixation requirement because the audiovisual display of the video game was in a medium that could be perceived directly).

142 17 U.S.C. §102(a) states "[c]opyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression." *See* Baker v. Selden, 101 U.S. 99, 100 (1879), *abrogated by* 17 U.S.C. § 102 (2010).

143 Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 351, 355 (1991); *see also* Sega Enters. v. Accolade, Inc., 977 F.2d 1510, 1519–20 (9th Cir. 1992).

144 Shaeffer, *supra* note 11, at 346, 371; *see also* Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 838 (Fed. Cir. 1992); H.R. REP. 94–1476, at 51, (1976) (indicating Congress intended to protect original works and amended the Copyright Act in light of the emerging technology as to not hinder new ideas while balancing creativity with promotion of learning and culture).

145 1 MELVILLE B. NIMMER & DAVID NIMMER ON COPYRIGHT § 2.01[A][1] (2017), Lexis (database updated Nov. 2017); *see, e.g.*, Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 830, 837 (10th Cir. 1993) (using the program "to determine the belt size" was original because the plaintiff made the program and there were various ways to create the program).

146 *Feist Publ'ns, Inc.*, 499 U.S. at 345, 348 (stating even though a work can be copyrighted, that does not mean every part of the work is protected because originality "remains the *sine qua non* of copyright;" therefore, copyrightability extends only to those aspects that are "original to the author."); *see also* Mislow, *supra* note 12, at 757–58 (explaining how the work that is original has to be a "work of authorship," which means any work that falls into 17 U.S.C. §102(a)); L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486, 490 (2d Cir. 1976) (explaining originality does not require novelty or ingeniousness).

identical to a previous or another version, as long as the creator "independently creates" the latter works from its previous work, the subsequent work may not infringe the previous work.[147] For instance, if two programmers create the same program unbeknownst to them, both programs are copyrightable because each program was "independently created."[148] The required level of creativity is "extremely low" that "a slight amount will suffice."[149]

2. Fixation

Fixation means the work is in "a tangible medium of expression" and is "sufficiently permanent or stable to permit it to be perceived or reproduced."[150] For example, a live performance of a song or an image is not fixed, but a program is fixed.[151] The difference between those works is that a program is in tangible form and is "sufficiently permanent" or can be reproduced.[152] Usually, plaintiffs can prove originality and fixation by showing to the court he or she registered the work with the Copyright Office.[153]

---

147 *Feist Publ'ns, Inc.*, 499 U.S. at 345; *see also* Hi–Tech Video Prods. v. Capital Cities/ABC, Inc., 58 F.3d 1093, 1095 (6th Cir. 1995) (presuming plaintiff's work was copyrighted because plaintiff had a certificate of registration); Stephen Preonas, *Mergercide, When Good Copyrights Go Bad: A Recommendation for a Market-Based, Defendant-Centric Approach to the Merger Doctrine in the Context of Complications*, 11 INTELL. PROP. L. BULL. 89, 90 (2006) (explaining how two works required very little creativity to quality for originality).

148 *See* L. Batlin & Son, Inc., 536 F.2d at 490; Velasco, *supra* note 5, at 256.

149 *Feist Publ'ns, Inc.*, 499 U.S. at 345; *see* Harper & Row Publishers, Inc. v. Nation Enterprises, 471 U.S. 539, 548, (1985) (providing one may use another author's work without infringing on the original author only if it does not "unfairly appropriate" the copied work). *But see* Russ VerSteeg, *Sparks in the Tinderbox: Feist, Creativity, and the Legislative History of the 1976 Copyright Act*, 56 U. PITT. L. REV. 549, 552, 556 (1995) (explaining how the Congress and the courts have not defined creativity).

150 17 U.S.C. § 101 (2010); *see also* S. REP. NO. 94-473, at 51-52 (1975) (stating the copyright law views a work as fixed when material objects can be embodied or is in tangible form); 1 NIMMER & NIMMER, *supra* note 145, at § 2.03[B][1] (defining fixation to mean that "the work as fixed can be perceived either directly or with the aid of a machine or other device."); *see, e.g.*, Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1247 (3d Cir. 1983) (explaining how the ROM was fixed in a tangible medium because it was capable of repeating its features).

151 *See* Megan Carpenter & Steven Hetcher, *Function over Form: Bringing the Fixation Requirement into the Modern Era*, 82 FORDHAM L. REV. 2221, 2225 (2014) (explaining how copyright law fails to protect contemporary art that is not fixed because it cannot be perceived or reproduced); *accord* Cmty. for Creative Non–Violence v. Reid, 490 U.S. 730, 737 (1989) (noting the programmer who fixed in a "tangible medium" such software will be the copyright owner).

152 1 NIMMER & NIMMER, *supra* note 145, at § 2.03[B][2] (quoting 17 U.S.C. § 101 (2010)); *see, e.g.*, Williams Elecs., Inc., v. Artic Int'l, Inc., 685 F.2d 870, 874 (3d Cir. 1982) (explaining the Defender game was fixed because it repeated the audiovisual features).

153 17 U.S.C. § 408(a) (2010); 4 NIMMER & NIMMER, *supra* note 48, at § 13.01[A]; *see also* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1492 (D. Minn. 1985) (holding

C. Limitations on the Copyright Scope for Computer Programs

Although a program may meet the requirements of originality and fixation, courts may limit certain aspects of a program for copyright protection.[154] Under the Copyright Act, a programmer's work does not have protection to his or her ideas, but rather to the expression of those ideas.[155] In other words, copyright protection does not extend to the main function of the work.[156] As a consequence, courts rely on the merger doctrine and employ other tests to determine what parts of the program are protected.[157]

When a court examines what aspects of a program are protected, it compares aspects of the program that are the expressions, such as the source code or algorithm, from parts of the program that are the idea, or what the computer performs for the user.[158] While an individual could not copy the programmer's code, one could produce the same end result or namely, the idea itself.[159] All programs have literal and non-literal aspects.[160] The literal aspects of a program are the source,

---

the plaintiff met the requirement of originality because it received the certification of registration from the Copyright Office of the ROM).

154 Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 834 (10th Cir. 1993); *see also* Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1476 (9th Cir. 1992) (providing courts should also look to the scope of the alleged copyrighted material).

155 17 U.S.C. § 102(b) (2010); *see, e.g.*, Tetris Holding, LLC v. XIO Interactive, Inc., 863 F. Supp. 2d 394, 409 (D.N.J. 2012) (holding the video game did not have copyrightability because the general ideas could not be protected).

156 *Gates Rubber Co.*, 9 F.3d at 833; *see also* Martyniuk, *supra* note 11, at 1342 (explaining how courts have been separating "the protected portions of the work from the unprotected portions" to determine how much copyright protection the work should receive).

157 Mazer v. Stein, 347 U.S. 201, 217 (1954); Baker v. Selden, 101 U.S. 99, 100 (1879), *abrogated by* 17 U.S.C. § 102 (2010). The idea-expression states copyright law does not afford "protection against the use of underlying ideas if they are expressed in another format." VERGARI & SHUE, *supra* note 12, at 521. The merger doctrine states, "when there is only one way, or a very few ways, to express an idea, the expression is said to merger with its idea and is not protected." Velasco, *supra* note 5, at 252–54, 256 (stating the other test includes *scènes à faire* and useful article doctrine).

158 Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 703 (2d Cir. 1992); *see also* William F. Patry, *Copyright and Computer Programs: It's All in the Definition*, 14 CARDOZO ARTS & ENT. L.J. 1, 33 (1996) (explaining how the protections of "an original work of authorship" do not extend to the ideas, which originates from section 102(b) of the Copyright Act). The source code is "[t]he literal text comprising the program's instructions." Ogilvie, *supra* note 61. An algorithm as "[a] step-by-step procedure, or defined set of instructions, designed to solve a particular problem or produce a particular result." E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1487 (D. Minn. 1985).

159 Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1249 (3d Cir. 1993); Conley & Bryan, *supra* note 109, at 571.

160 *Comput. Assocs. Int'l*, 982 F.2d at 701; Jeffrey Malkan, *Rule–Based Expression in Copyright Law*, 57 BUFF. L. REV. 433, 451 (2009).

object code, and the flowchart, and the non-literal aspects are the structure, sequence, organization, and user interface.[161] While literal parts of a program are protected if original,[162] non-literal parts may be protected or unprotected because the non-literal aspects are functional in nature and thus can be unprotected.[163] Establishing which part of a program is an idea or an expression is an issue attorneys frequently litigate.[164]

1. The Idea-Expression Line

To rectify the troublesome nature of determining the copyright protection of non-literal elements of computer programs, the Third Circuit in *Whelan Assocs. v. Jaslow Dental Lab., Inc.* described the line between the idea and the expression of an idea and stated, "the purpose or function of a utilitarian work would be the work's [idea] and everything that is not necessary to that purpose or function would be part of the expression of the idea."[165] In other words, if a program tells a user the distance and time to his or her desired destination, which is the idea of the program and is unprotected, any part of the program that is not necessary to informing the user about his or her destination, such as the source code

---

161 Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 45–46, 54 (D. Mass. 1990) (providing flowcharts are commonly utilized during early stages of programming, and the user interface, a non-literal aspect of the program, was protected); 1 NIMMER & NIMMER, *supra* note 145, at § 2A.10[B] (defining the object code as a "machine–readable language."); *see also* Menell, *supra* note 16, at 1048 (explaining how various courts have held the literal parts of a program are the source and object code).

162 Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 (3d Cir. 1986); Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1355 (Fed. Cir. 2014).

163 *Comput. Assocs. Int'l*, 982 F.2d at 696; Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175 (9th Cir. 1989); *see also* Stephen H. Eland, *The Abstraction-Filtration Test: Determining Non-Literal Copyright Protection for Software*, 39 VIL. L. REV. 665, 667, 670 (1994) ("[D]efining the scope of copyright protection for software" has become difficult because "computer programs are utilitarian in nature, and computer technology advances at a rapid rate." Developers have become concerned about the program's non-literal aspects receiving protection).

164 *Comput. Assocs. Int'l*, 982 F.2d at 705; *see also* Maiorana, *supra* note 13, at 152 (affording protection to non-literal aspects of program "has varied in recent years from board coverage to practically no coverage at all."); *see, e.g.*, Paycom Payroll, LLC v. Richison, 758 F.3d 1198, 1204 (10th Cir. 2014) (determining whether the defendant's program took "too much" of the plaintiff's program and must be tested).

165 *Whelan Assocs.*, 797 F.2d at 1236. The idea-expression line famously originated in Baker v. Selden, 101 U.S. 99, 102, 105 (1879), *abrogated by* 17 U.S.C. § 102 (2010); *see also* Sid & Marty Krofft Television v. McDonald's Corp., 562 F.2d 1157, 1168 (9th Cir. 1977), *superseded by* 17 U.S.C. §504(b) (defining the idea-expression line as when the idea and the expression coincide and "the expression provide[s] nothing new or additional over the idea," the work has protection); *see, e.g.*, Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240,1253 (3d Cir. 1993) (explaining that if programmers cannot create programs that perform the same function as Apple's operating system program, then the operating system is unprotected).

and the algorithm, is protected.[166] The Court reasoned while the literal aspects of a program have copyright protection, the non-literal, the structure, sequence, or organization, have protection depends on whether those aspects are not essential to the purpose of the program.[167] However, courts struggle with separating the idea from the expression because the programmer has to create the flowchart, the expression, to produce the main function of the program, the idea.[168] Furthermore, courts have heavily criticized this test and instead apply other tests to determine what is the original expression and what parts are protectable and unprotectable of a program.[169]

2. The Merger Doctrine

When elements of a program include both the idea and the expression of the idea, courts apply the merger doctrine.[170] This doctrine states, "when there are a limited number of ways to express an idea, the idea is said to 'merge' with its expression, and the expression becomes unprotected."[171] When the defendant admits to copying some portions of the programmer's work, the defendant can use

166 *See Whelan Assocs.*, 797 F.2d at 1236; *see also* Clapes et al., *supra* note 64, at 1552 (stating that programs have protection if programmers can describe their program in various ways).

167 *Whelan Assocs.*, 797 F.2d at 1248; Reger, *supra* note 73, at 225; *see also* Miller, *supra* note 11, at 995 (explaining how courts have provided programmers with copyright protection to other aspects of a computer program, such as data structures that formats and stores data and modules that perform tasks and are pieces that constitute the program).

168 Shaeffer, *supra* note 11, at 346, 368; *Paycom Payroll, LLC*, 758 F.3d at 1205; *see* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (finding the court struggled to find the line between the idea and the expression of a work); *see also* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 45 (D. Mass. 1990) (explaining courts consider flowchart to be protected, "if sufficiently detailed and original.").

169 Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522, 539 (6th Cir. 2004); *Lotus*, 49 F.3d at 814; *Comput. Assocs. Int'l*, 982 F.2d at 705 (applying the abstraction-filtration-comparison test, which breaks down each aspect of the program); Plains Cotton Cooperative Assocs. of Lubbock, Texas v. Goodpasture Compt. Serv., Inc., 807 F.2d 1256, 1262 (5th Cir. 1987).

170 *Comput. Assocs. Int'l*, 982 F.2d at 708; *see also* Englund, *supra* note 62, at 902 (explaining whether the merger doctrine applies with modules depends on "the use of this particular set modules is necessary efficiently to implement that part of the program's process that is implemented in the common client module."); *see, e.g.*, CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc., 44 F 3d. 61, 64 (2d Cir. 1994) (stating the Red Book valuations merged with the entry).

171 Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1359 (Fed. Cir. 2014). The merger doctrine applies when the expression of an idea can be stated in only one way and copyright law does not protection the expression. Velasco, *supra* note 5, at 254; *see also* Morrissey v. Proctor & Gamble Co., 379 F.2d 675, 678 (1st Cir. 1967) (stating a broad view of the merger doctrine: when there is one way to express an idea, allowing "copyrighting would mean that a party or parties . . . could exhaust all possibilities of future use of the substance."); Green,

the merger doctrine as a defense to copying, by proving the alternatives to creating the protected aspects are inefficient and the only reasonable method is to copy the program.[172] For instance, if the main function of a program merges with the code, which is usually the case, and the court finds the code is necessary to developing the program's function, it will hold the code is unprotected and rule in favor of the defendant.[173]

When a court inquires into whether the protected elements of a program intertwine with the unprotected ones, it must be prudent because creating the program's definition depends on the process to produce the program, such as the source code and the algorithm.[174] Courts apply the merger doctrine frequently to copyright infringement claims because they want to incentivize competition and promote efficiency.[175] Since courts view the functionality of a program from the user's perspective, they fail to understand programming is more than a simple process involving the development and writing of a program but requires years

supra note 126 (explaining how Compt. Assocs. Int'l v. Altai, Inc. provides a broad view of the merger doctrine; therefore, misunderstanding the programming process of an experienced programmer).

172 ATC Distrib. Grp., Inc. v. Whatever It Takes Transmission & Parts, Inc., 402 F.3d 700, 709 (6th Cir. 2005); *see* Timothy S. Teter, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 STAN. L. REV. 1061, 1074 (1993) (applying the merger doctrine depends on how courts define the programmer's idea or the main function of the program).

173 Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240,1253 (3d Cir. 1993); *see, e.g.*, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524, 1530, 1532 (9th Cir. 1992) (explaining how the court ruled in favor of defendants when court found code necessary to the function of the program).

174 Shaeffer, *supra* note 11, at 368; Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 838 (10th Cir. 1993); *see also Apple Comput.*, 714 F.2d at 1253 (reasoning the defendant had alternative ways of expressing the translation of a source code into an object code). An algorithm is a "step-by-step formula for accomplishing a task." SCHNEIDER ET AL., *supra* note 2, at 3; *see, e.g., Lexmark Int'l, Inc.*, 387 F.3d at 523 (holding Lexmark's "lock-out" code of the program merged with the code sequence; therefore, the code sequence was unprotected).

175 *Comput. Assocs. Int'l*, 982 F.2d at 708; Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971); Pamela Samuelson, *Reconceptualizing Copyright's Merger Doctrine*, 63 J. COPYRIGHT SOC'Y U.S.A. 417, 428 (2016); *cf.* 4 NIMMER & NIMMER, *supra* note 48, at § 13.01[B] (criticizing Comput. Assocs. Int'l v. Altai, Inc. for misstating the purpose of the merger doctrine, which is concerned only with efficiency). Another judicially-created doctrine is *scènes à faire*, which states "those expressions that are standard, stock, or common to a particular topic or that necessarily follow from a common theme or setting." *Gates Rubber Co.*, 9 F.3d at 838.

of experience and knowledge to produce an intelligent and functioning program.[176] Consequently, Congress should not deny a programmer's work copyright protection.[177]

III. COPYRIGHT INFRINGEMENT

Once the court determines the programmer's work has originality and fixation and is within the copyright limitations, he or she may have a cause of action for copyright infringement against the defendant for unauthorized copying.[178] For a copyright infringement claim, the programmer must prove "ownership of a valid copyright" and the "copying of constituting elements of the work that are original."[179]

A. Ownership of a Valid Copyright

The programmer can usually demonstrate "ownership of a valid copyright" by presenting to the court the certificate of registration.[180] Once the programmer proves he or she has "a valid copyright," the programmer must show the defendant "unlawfully appropriated protection portions of the copyrighted work."[181]

---

176 Shaeffer, *supra* note 11, at 370; *see also* Faust, *supra* note 127 (explaining the merger doctrine "undermines" the Constitutional intent of copyright law, which is to provide some protection, and courts appear to do that "by denying protection to unprotectable elements."); Menell, *supra* note 16, at 1101 (describing programming as a human learning process and courts should be cognizant of the importance features that programmer created). *See generally* VERGARI & SHUE, *supra* note 12, at 510–11 (asserting technology is "undermining many of the mechanisms that governed the system in the past" and the difficult role technology has in the realm of intellectual property and law).

177 Shaeffer, *supra* note 11, at 370; *see* Aufrichtig, *supra* note 15, at 136–37 (explaining the "wide and growing importance" of computer programs in society and how one has to use a program to continue one's pursuit of life).

178 17 U.S.C. § 501 (2010); *see also* 17 U.S.C. § 106 (2010) (granting the author five exclusive rights, such as the right to reproduce, to distribute, and to perform); *see, e.g.*, Montgomery v. Noga, 168 F.2d 1282, 1293 (11th Cir. 1999) (holding the defendant was liable for copyright infringement because it violated "an exclusive right" of or copied the plaintiff's copyrighted program and used it without permission).

179 28 U.S.C. § 1338(a) (2011) (filing a copyright infringement claim in federal court because the federal courts have exclusive jurisdiction over disputes arising under the Copyright Act); Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 361, 359 (1991); *see also* Arenault, *supra* note 9, at 139 (explaining how copyright infringement is a "'clean room' method of copying" programs).

180 *Gates Rubber Co.*, 9 F.3d at 832; 17 U.S.C. § 410(c) (2010). The requirements of registration are an application of registration, a deposit of the work, and a fee. 17 U.S.C. §§ 408(b), 409, 708(a); U.S. COPYRIGHT OFFICE, *supra* note 3, at § 1107.1.

181 *Gates Rubber Co.*, 9 F.3d at 831–32; *see also* Dunn, *supra* note 55 (finding that copyrighting is more complicated that originally thought).

Once the court decides the programmer has met the presumption Section 410(c) of the Copyright Act, the defendant has the burden to overcome it.[182]

B. Copying Elements of the Original Work

To prove the defendant copied the program, the programmer has to show the defendant copied original aspects of the program.[183] First, the court conducts a factual inquiry as to whether the defendant copied aspects of the programmer's program.[184] Then, the court determines whether the defendant's copying of those elements of the program were substantially similar to the programmer's work because not all copying is copyright infringement.[185]

1. Factual Copying

To determine factual copying, the programmer must show the defendant had the opportunity to view or access the program.[186] The programmer can provide direct proof to the court.[187] However, proving the defendant had physical access

---

182 *Gates Rubber Co.*, 9 F.3d at 832; *see, e.g.*, Williams Elecs., Inc., v. Artic Int'l, Inc., 685 F.2d 870, 873 (3d Cir. 1982) (stating since the plaintiff had the certificates of registration, which constituted the "prima facie evidence of the validity of the plaintiff's copyright.").

183 *Feist Publ'ns, Inc.*, 499 U.S. at 361–62. To prove copying, the plaintiff has to show the defendant factually copied and the defendant's work is substantially similar to the plaintiff's work. Runstadler Studios, Inc. v. MCM Ltd. P'ship, 768 F. Supp. 1292, 1296 (N.D. Ill. 1991); *see also* Alan Latman, *Probative Similarity as Proof of Copying: Toward Dispelling Some Myths in Copyright Infringement*, 90 COLUM. L. REV. 1181, 1189 (1990) (explaining how proof of copyright protection and no other exclusions apply before the court can entertain a claim of infringement).

184 *Gates Rubber Co.*, 9 F.3d at 832; *see, e.g.*, Castle Rock Entm't v. Carol Publ'g Grp., Inc., 955 F. Supp. 260, 264 (S.D.N.Y. 1997) (holding the defendant's book factually copied information and dialogues of the plaintiff's television show because the book is based on the show).

185 Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946); *see, e.g.*, Wallace Compt. Serv. v. Adams Bus. Forms, 837 F. Supp. 1413 (N.D. Ill. 1993) (finding substantial similarity because, by comparing the plaintiff's telephone on a photograph to the defendant's telephone, the court determined how both parties' telephone booths had their message book at a similar angle).

186 Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 838 (Fed. Cir. 1992); *see* 4 NIMMER & NIMMER, *supra* note 48, at § 13.01[B] (describing the inquiry as factual and is not concerned with the legal question of whether the copying was substantial).

187 *Atari*, 975 F.2d at 838; *see also* McIntosh v. N. California Uni. Enter. Co., 670 F. Supp. 2d 1069, 1087 (E.D. Cal. 2009) (explaining whether defendant copied as a factual matter is an issue of fact, and the defendant asserted the plaintiff lacked evidence of direct proof); VERGARI & SHUE, *supra* note 12, at 553 (having physical access can arise from the defendant's "past or present unauthorized possession of the original program's source codes, the existence of the defendant's copies of the original program after contract termination.").

to the programmer's work is difficult.[188] Instead, the programmer can demonstrate access through indirect proof by arguing the defendant had a "reasonable opportunity" to appropriate the programmer's work.[189]

2. Substantial Similarity

After the programmer establishes factual copying, the court compares the programmer's work with the defendant's work to must determine whether the copying was substantial because the defendant must have committed an unlawful copying.[190] The court must determine what parts of the program are protected because copyright law provides protection to those parts that are protectable, unless the protectable parts merge with the unprotectable parts.[191] Most courts have provided the literal aspects of a computer program, such as the object code, with protection; however, they struggle determining what non-literal aspects, such as subroutines, are protected.[192] As a result, courts have developed various

---

188 Johnson v. Gordon, 409 F.3d 12, 18 (1st Cir. 2005); Dunn, *supra* note 55, at 509; Olufunmilayo B. Arewa, *The Freedom to Copy: Copyright, Creation, and Context*, 41 U.C. Davis 477, 486 (2007); *see also* Trek Leasing, Inc. v. United States, 66 Fed. Cl. 8, 11 (Fed. Cl. 2005) (admitting to having access and "factually copied" the plaintiff's work, demonstrating defendant usually do not contest this element but rather the substantial similarity element).

189 4 NIMMER & NIMMER, *supra* note 48, at § 13.02[C]. The defendant can rebut that assertion by bringing "evidence of independent creation to rebut the inference of copying created by the evidence of access and factual similarity." *Gates Rubber Co.*, 9 F.3d at 833 n.8.

190 Feist Publ'ns, Inc. v. Rural Tel. Servs. Co., Inc., 499 U.S. 340, 361 (1991); *see, e.g.*, Close To My Heart, Inc. v. Enthusiast Media LLC, 508 F. Supp. 2d 963, 968 (D. Utah 2007) (holding whatever substantial similarity existed between craft company's proprietary scrapbooking page patterns and competitor's SIP emerged from unprotected materials and, therefore, the company failed to establish a likelihood of success on its claim for copyright infringement); Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1341 (5th Cir. 1994) (arguing the defendant copied the program's user formats; therefore, the copying is actionable).

191 *Gates Rubber Co.*, 9 F.3d at 832; *see* 17 U.S.C. § 102(b) (2010) (stating the ideas, process, system, method of operation, and others do not have copyright protection). When the expression of the program, such as the object code, cannot be separated from the idea of the program, they have merged and the object code, which normally has copyright protection, will have no protection. Apple Comput., Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1021 (N.D. Cal. 1983).

192 Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1355 (Fed. Cir. 2014) (providing protection to "the declaring code and the structure, sequence, and organization of the 37 Java API packages."); Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175 (9th Cir. 1989) (stating the non-literal aspects of a program are the structure, sequence, organization, and user interface); Apple Comput. v. Franklin Comput. Corp., 714 F.2d 1240, 1249 (3d Cir. 1983); Velasco, *supra* note 5, at 259. An object code is a "machine-readable code," which the computer executes directly to produce the user's task. A subroutine is a "set of instructions that perform a specific computational procedure whenever called on to do so." VERGARI & SHUE, *supra* note 12, at 554–55.

tests to determine the extent to which the programmer's work has copyright protection.[193]

i. The Abstraction-Filtration-Comparison Test

In *Computer Associates, Inc. v. Altai, Inc*., the Second Circuit created the abstraction-filtration-comparison ("AFC") test to determine whether there was "substantial similarity" between the original program and the infringing program and whether the nonliteral aspects of the program would be protected.[194] Courts primarily employ the AFC test for computer programs because the test will leave the programmer with the protectable elements and distinguishes effectively the idea from the expression of the idea for programs.[195]

Under the abstraction step, the court "dissects the allegedly copied program" to separate each aspect of the program and ascertains whether each aspect of the program is an idea, a process or a method.[196] The court divides the program into

193 *Gates Rubber Co.*, 9 F.3d at 833; *see also Johnson Controls, Inc.*, 886 F.2d at 1175 (explaining whether a non-literal aspect has protection depends on whether it part of the idea or the expression of an idea); McGahn, *supra* note 12, at 113 (employing tests, such as the ordinary observer, the extrinsic/intrinsic, and the total concept and feel, to determine the copyright protection of a program).

194 *Gates Rubber Co.*, 9 F.3d at 834; Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706 (2d Cir. 1992); *see also* Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807, 815 (1st Cir. 1995) (using the AFC test only for determining the nonliteral aspects of a program because the court has to ascertain whether the program "as a whole is copyrighted.").

195 Comput. Mgmt. Assistance Co. v. Robert F. DeCastro, Inc., 220 F.3d 396, 401 (5th Cir. 2000); *Gates Rubber Co.*, 9 F.3d at 834–39; *Comput. Assocs. Int'l*, 982 F.2d at 715 (affirming the AFC test and explaining the procedure for its application); Faust, *supra* note 127, at 140–41 (explaining how the court in Gates Rubber Co. v. Bando Chem. Indus. adopted AFC test principally for computer programs with copyright infringement cases). Before the test in Comput. Assocs. Int'l v. Altai, Inc. gained prominence, courts utilized various tests with the copyright scope of computer programs, specifically for the non-literal aspects: the iterative, the ordinary observer, the total concept and feel, and the intrinsic-extrinsic. Mark E. Dailey, *Abstraction, Filtration, Comparison: The Difficult Task of Defining and Applying an Appropriate Substantial Similarity Test for Computer Software*, 34 SUFFOLK U.L. REV. 415, 429-31 (2001).

196 Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1343 (5th Cir. 1994); *Comput. Assocs. Int'l*, 982 F.2d at 707; *see also* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (describing the Judge Learned Hand's abstractions test, which was developed for plays and to isolate the "what the [work] is about" from the artistic elements).

six levels: the main function,[197] the source code,[198] object code,[199] modules,[200] algorithms,[201] and data structures,[202] and the program structure or architecture.[203] The court retraces and maps each step the programmer took to create the program because it must identify the main purpose of the program with specificity without referencing the technical elements, data structures and modules, to ensure the programmer's protected aspects remain protected.[204] The purpose of this step is to separate the ideas of the program from the expressions of the idea.[205]

In the filtration phase, the court examines the protectable aspects of the program to determine whether that aspect is an idea, those listed in section 102(b) of the Copyright Act,[206] or whether those protected aspects are "dictated by considerations of efficiency," the application of judicially-created defense to copyright infringement, the merger doctrine.[207] Once the court identifies each aspect

---

197 The main function of a program is the task that the computer performs and produces for the user. VERGARI & SHUE, *supra* note 12, at 2.

198 A source code is a program that is written in "a human-readable computer language." VERGARI & SHUE, *supra* note 12, at 555.

199 An object code is a binary code, consisting of zeros and ones, "that directs the computer to perform a function." Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230–31 (3d Cir. 1986).

200 A program consists of modules, which perform a function. HUGHES & MICHTOM, *supra* note 55, at 22–23.

201 An algorithm is a procedure for "a procedure for solving a given type of mathematical problem." Michael Gemignani, *Should Algorithms Be Patentable*, 22 JURIMETRICS J. 326, 327 (1982).

202 A data structure is "an organization of data" that holds data for which the algorithm can call on and have the computer execute those algorithms. Hamilton & Sabety, *supra* note 50.

203 Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993) (defining "[t]he program's architecture or structure is a description of how the program operates in terms of its carious functions, which are performed by discrete modules, and how each of these modules interact with each other."); Comput. Assocs. Int'l v. Altai, Inc., Inc., 982 F.2d 693, 707 (2d Cir. 1992) (providing a detailed "anatomical guide" to separating each part of the program); *see also* Matthew P. Larvick, *Questioning the Necessity of Copyright Protection for Software Interfaces*, 1994 U. Ill. L. Rev. 187, 196 (1994) (describing a program's abstraction level is from "individual lines of codes" to "higher-level modules.").

204 *Gates Rubber Co.*, 9 F.3d at 835; *Comput. Assocs. Int'l*, 982 F.2d at 707.

205 *Comput. Assocs. Int'l*, 982 F.2d at 706; *see* Mark A. Lemly, *Convergence in the Law of Software Copyright*, 10 BERKELEY TECH. L.J. 1, 32 (1995) (arguing the abstraction step "should be used only as an analytic guide" because it is difficult "to reconcile" Lotus Dev. Corp. v. Borland Int'l, Inc. with Comput. Assocs. Int'l v. Altai, Inc.); *see also* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (equating the idea of abstraction to that of the tension between plagiarist and a playwright's play, there becomes a point where general statements from a play are no longer protected because a playwright could prevent the expression of general ideas).

206 17 U.S.C. § 102(b) (2010) (stating "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described.").

207 *Comput. Assocs. Int'l*, 982 F.2d at 707, 710-11; *see also* Faust, *supra* note 127, at 142

as protectable or unprotectable, it disregards the unprotectable aspects.[208] Then, the court deals with the protected aspects of the program and uses them to learn whether the defendant substantially copied them because copyright law requires actionable copying.[209]

Once the court sifts through all of the aspects of the program, in the comparison stage, the court compares the protected aspects of a programmer's work with the defendant's program and evaluates to learn whether the defendant substantially copied the programmer's work.[210] After the court determines the programmer's work has copyright protection and he or she has a claim for copyright infringement, the court may not allow the programmer to enforce his or her legal rights because he or she may have to wait four or five months for the Copyright Office to register the programmer's work.[211]

IV. REGISTRATION

Pursuant to section 411(a) of the Copyright Act, the programmer must register his or her work for copyright with the Copyright Office before filing a lawsuit.[212]

(defining the merger doctrine as "when the idea intertwines with the expression such that it is impossible to separate them, the expression is said to have 'merged' with the idea."). In addition to the merger doctrine, the court filters out aspects by using the *scènes à faire* or public domain. Larvick, *supra* note 203, at 197.

208 *Comput. Assocs. Int'l*, 982 F.2d at 707; Concrete Mach. Co., Inc. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 606 (1st Cir. 1988) (explaining "[w]hen the idea and its expression are not completely inseparable, there may still be only a limited number of ways of expressing the idea."); Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp., 562 F.2d 1157, 1167 (9th Cir. 1977) (stating how "near identity may be required in some cases not because the expression of those works are things, but because the expression of those works and the idea of those works are indistinguishable.").

209 Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1341 (5th Cir. 1994); *Gates Rubber Co.*, 9 F.3d at 831–32, 836; *Comput. Assocs. Int'l*, 982 F.2d at 710; *see* Patry, *supra* note 158, at 53 (asserting the AFC test is fixed on the literal elements of the program and does not focus on the nature of "programs as a set of statements or instructions.").

210 *Comput. Assocs. Int'l*, 982 F.2d at 710; Larvick, *supra* note 203, at 199 (describing the AFC test as striving to meet various goals, such as incentivizing innovation and competition, protecting the interest of the programmer and the consumer, and maintaining an efficient copyright system); *see also Gates Rubber Co.*, 9 F.3d at 836 (explaining, after the AFC test, the merger doctrine may apply because the literal elements, which are always protected, may merge with the protected nonliteral parts).

211 Metro. Reg'l Info. Sys. V. Amer. Home Realty Net., 722 F.3d 591, 596, 599 (4th Cir. 2013); Casa Dimitri Corp. v. Invicta Watch Co. of Am., 270 F. Supp. 3d 1340, 1348-49 (S.D. Fla. 2017); Bracey, *supra* note 24, at 120.

212 17 U.S.C. § 411(a) (2010). The benefits of registration include attorney fees and statutory damages. 17 U.S.C. § 412 (2010); *see also* Hogan, *supra* note 23, at 849 (explaining the purpose of creating a federal registration system and having the Copyright Office, which is at the Library of Congress, is to collect literary works because Congress wanted to promote human creativity and knowledge). *See generally* 3 MELVILLE B. NIMMER & DAVID NIMMER ON COPYRIGHT § 12.08 (2017), Lexis (database updated Nov. 2017).

The Copyright Act defines registration as "a registration of a claim in the original or the renewed and extended term of copyright."[213] Section 410(d) states, "[t]he effective date of a copyright registration is the day on which an application, deposit, and fee, which are later determined by the Register of Copyrights or by a court of competent jurisdiction to be acceptable for registration, have all been received in the Copyright Office."[214] However, this section of the Copyright Act fails to state whether the Copyright Office considers a programmer's work registered upon filing an application for registration or when the applicant receives a decision on his or her registration from the Copyright Office.[215] Courts that choose the former approach will allow the claim, such as copyright infringement, to proceed and the ones that choose the latter approach will dismiss the lawsuit based on lack of subject-matter jurisdiction, creating a circuit split and leaving the programmer with minimal protection.[216] Registration is only a prerequisite for a lawsuit, and courts decide the validity of the copyright.[217]

Pursuant to the holding of *Reed Elsevier, Inc. v. Muchnick*, registration is not a prerequisite for subject-matter jurisdiction.[218] However, courts can still dismiss

---

213 17 U.S.C. § 101 (2010).

214 17 U.S.C. § 410(d) (2010). *See generally* 17 U.S.C. § 408(b) (2010) (specifying the requirements for a deposit); 17 U.S.C. § 409 (2010) (listing the application materials in addition to other materials as required by the Copyright Office); 17 U.S.C. § 708 (2010) (requiring a fee for filing the registration application); U.S. COPYRIGHT OFFICE, *supra* note 3, at §§ 504.2, 607 (requiring the programmer to deposit the source code, which is a human-readable language, or a "set of statements and instructions authored using a programming language," to the Copyright Office of a computer program in general and, for specific types of programs, the Copyright Office has additional requirements).

215 17 U.S.C. § 410(d) (2010); 17 U.S.C. § 411(a) (2010); *see also* Thomas M. Landrigan, *Application or Registration?: Confusion Regarding the Copyright Act's Prerequisite to Copyright Infringement Lawsuits*, 44 L. Rev. 581, 581 (2011) (explaining the differences between the two views and the "lack of clarity" among federal judges).

216 *Compare* Cosmetic Ideas, Inc. v. IAC/InteractiveCorp, 606 F.3d 612, 621 (9th Cir. 2010) (finding the plaintiff's copyright infringement suit could proceed because the Copyright Office received it); Apple Barrel Prods. v. Beard, 730 F.2d 384, 386 (5th Cir. 1984) (holding the plaintiff met the requirement of registration upon filing the application) *with* La Resolana Architects v. Clay Realtors Angel Fire, 416 F.3d 1195, 1202 (10th Cir. 2005) (holding a copyright was not "registered" until Copyright Office actually approved or rejected application), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010); M.G.B. Homes, Inc. v. Ameron Homes, Inc., 903 F.2d 1486, 1489 (11th Cir. 1990) (dismissing the plaintiff's copyright infringement suit because the Copyright Office did not make a determination on the application), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

217 Ward v. Nat'l Geographic Soc., F. Supp. 2d 429, 444-45 (S.D.N.Y. 2002); Conley & Bryan, *supra* note 109, at 578; *see also* Bracey, *supra* note 21, at 121 (explaining how if the Copyright Office denies registration, the plaintiff can contest it by filing a suit under the Administrative Procedure Act or the Copyright Act). *But see* Cain, *supra* note 22, at 384 (explaining how the Copyright Office could intervene in the infringement suit and argue the plaintiff's work does not have copyrightability).

218 Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154, 168 (2010); *see also* Positive Black

a claim for copyright infringement on other grounds.[219] As a result, there is still a split in the circuits[220] as to when a work is registered – after the application has been filed (the application approach) or after the Copyright Office has received the application and made a determination (the registration approach).[221]

A. Registration Approach

Under this approach, the Tenth and Eleventh Circuits have held a program is considered registered once the Copyright Office has made a determination on the application; otherwise, they will dismiss his or her claim for not meeting the registration requirements.[222] However, it is unclear whether the registration approach has been met when the programmer receives the certificate of registration or when the Copyright Office sends a notification to the programmer about approving or rejecting his or her application.[223]

Talk, Inc. v. Cash Money Records, Inc., 394 F.3d 357, 365 (5th Cir. 2004), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

219 Zaslow v. Coleman, 103 F. Supp. 3d 657, 663 (E.D. Pa. 2015) (holding the plaintiff failed to state a claim because the Copyright Office determines the validity of their application; therefore, the plaintiff did not register their work); Kernel Records Oy v. Mosley, 694 F.3d 1294, 1311-12 (11th Cir. 2012) (showing a court can dismiss a claim based any of the available 12b defenses, except for subject–matter jurisdiction).

220 *Compare* Fourth Est. Pub. Benefit Corp. v. Wall–Street.com, LLC, 856 F.3d 1338, 1340 (11th Cir. 2017) (accepting registration upon receiving approval or denial) *with* Lakedreams v. Taylor, 932 F.2d 1103, 1108 (5th Cir. 1991) (adopting the application approach). Some circuits, such as the first, second, third, sixth, and eighth, have not ruled when registration has occurred. *See, e.g.*, Dawes-Lloyd v. Publish Am., LLLP, 441 Fed. Appx. 956, 957 (3d Cir. 2011); Action Tapes, Inc. v. Mattson, 462 F.3d 1010, 1014 (8th Cir. 2006) (adopting the registration approach); Johnson v. Gordon, 409 F.3d 12, 17 (1st Cir. 2005) (holding the certificate of registration is prima facie evidence of ownership and the validity of the copyright); Murray Hill Pubs., Inc. v. ABC Comms., Inc., 264 F.3d 622, 632 (6th Cir. 2001), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) (stating the court lacked subject-matter jurisdiction because the plaintiff failed to register the derivative song); Wales Indus. v. Hasbro Bradley, Inc., 612 F. Supp. 510, 511 (2d Cir. 1985) (finding no action for infringement of the copyright in any work shall be instituted until registration of the copyright claim has been made).

221 Prunté v. Universal Music Grp., 484 F. Supp. 2d 32, 40 (D.D.C. 2007) (adopting the application approach). *But see* Mays & Assocs. v. Euler, 370 F. Supp. 2d 362, 370 (D. Md. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) (adopting the registration approach). Seventh Circuit has chosen both approaches. *Compare* Chicago Bd. Of Educ. v. Substance, Inc., 354 F.3d 624, 631 (7th Cir. 2003) (filing an application meets registration requirement for a lawsuit) *with* Gaiman v. McFarlane, 360 F.3d 644, 655 (7th Cir. 2004) (requiring the plaintiff to receive decision on the application from the Copyright Office).

222 *Kernel Records*, 694 F.3d at 1302; *La Resolana Architects*, 416 F.3d at 1202; Cain, *supra* note 22, at 384; *see also* Bracey, *supra* note 23, at 127 (explaining how the registration approach relies on the text of the Copyright Act and the statutory interpretation).

223 2 NIMMER & NIMMER, *supra* note 22, at § 7.16[B][3][b][i]. *Compare* Strategy Source,

1. Pre-*Reed Elsevier, Inc. v. Muchnick*

Prior to *Reed Elsevier, Inc. v. Muchnick*, courts held the registration requirement was a "jurisdictional prerequisite" to a copyright infringement suit.[224] In *La Resolana Architects v. Clay Realtors Angel Fire*, 416 F.3d 1195 (10th Cir. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) the plaintiff argued the defendant had used its drawings for a townhouse project and filed suit on November 20, 2003.[225] The plaintiff applied for registration on November 8, 2003 and received confirmation from the Copyright Office about receiving its application materials, but the Copyright Office did not make a determination on the application.[226]

To determine the meaning of "registration," the Tenth Circuit first looked at the plain language of the Copyright Act and explained a plaintiff can initiate an infringement action only if the plaintiff has registered the work.[227] Both the applicant and the Copyright Office have to take action to meet the registration requirements.[228] The language of the statute does not suggest registration is met when the plaintiff receives confirmation from the Copyright Act of the application.[229] Instead, the Copyright Office has to ascertain the validity of the application.[230] Once the application is either rejected or accepted, the plaintiff can sue and the court would have subject-matter jurisdiction over the copyright infringement suit.[231]

Inc. v. Lee, 233 F. Supp. 2d 1, 3-4 (D.D.C. 2002), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) (holding the plaintiff must possess the certificate of registration to meet the requirements of 17 U.S.C. § 411(a)), *with La Resolana Architects*, 416 F.3d at 1208 (receiving a determination from the Copyright Office about the application suffices the registration requirement for a lawsuit).

224 *In re* Literary Works in Electronic Databases Copyright Litigation 509 F.3d 116, 117 (2d Cir. 2007), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010); Xoom, Inc. v. Imageline Inc., 323 F.3d 279, 282-83 (4th Cir. 2003), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010); Data Gen. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1163 (1st Cir. 1994), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

225 *La Resolana Architects*, 416 F.3d at 1197.

226 *Id.* at 1197–98.

227 *Id.* at 1200; *see also* Bracey, *supra* note 24, at 127 (explaining how the plain meaning of the Copyright Act supports the registration approach).

228 *La Resolana Architects*, 416 F.3d at 1200; *see* 17 U.S.C. §§ 408, 410 (requiring the applicant to file an application, submit a deposit of a copy of the work, pay a fee, examine the work, accept or deny registration, and issue a certificate of registration).

229 *La Resolana Architects*, 416 F.3d at 1200.

230 *Id.*; *see also* United States v. Beltran, 503 F.2d 1, 2 (1st Cir. 2007) (reasoning the defendant can rebut the Copyright Office's approval of registration).

231 *La Resolana Architects*, 416 F.3d at 1201; *see* Hogan, *supra* note 23 (criticizing the Tenth Circuit for misinterpreting the issue as "what registration requires, not whether it is required.").

Then, the Tenth Circuit looked at the interpretation of section 410(a).[232] This section also requires the Copyright Office to take affirmative steps, such as examining, registering, and then issuing the certificate of registration.[233] However, section 410(a) does not suggest only filing an application is sufficient to register a work.[234]

Once the Copyright Office approved the plaintiff's application, the issue was whether the plaintiff was required to have a certificate of registration.[235] The Tenth Circuit stated although the plaintiff may not have the certificate, the Copyright Office had notified the plaintiff about its determination of the plaintiff's application and he or she had met the registration requirement for litigation purposes.[236] The court's subject-matter jurisdiction depends on whether registration occurred and not on the issuance of the certificate.[237]

Finally, the Tenth Circuit provided the statute requires registration before filing a suit because Congress provided incentives and remedies to plaintiffs who registered.[238] The Tenth Circuit held it lacked subject-matter jurisdiction over the plaintiff's suit because the plaintiff did not register the drawings at the time the plaintiff filed the suit.[239]

---

232 *La Resolana Architects*, 416 F.3d at 1201; *see also* Strategy Source, Inc. v. Lee, 233 F. Supp. 2d 1, 3 (D.D.C. 2002), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) (interpreting 17 U.S.C. § 408(b) to conclude when the plaintiff sent the application and other required materials to the Copyright Office and the office made a determination, registration occurred).

233 *La Resolana Architects*, 416 F.3d at 1201; Loree Rodkin Mgmt. Corp. v. Ross-Simons, Inc., 315 F. Supp. 2d 1053, 1055 (C.D. Cal. 2004) (requiring the Copyright Office to "register a claim and issue a certificate after examining the deposited material and determining that it constitutes copyrightable subject matter.").

234 17 U.S.C. § 410(b) (2010); 17 U.S.C. § 408 (2010); *La Resolana Architects*, 416 F.3d at 1201; Mays & Assocs. v. Euler, 370 F. Supp. 2d 362, 368 (D. Md. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010) (station the reason why 17 U.S.C. §§ 410, 411 has the word "application" separate from the "registration," is because having filing and registration be synonymous would render the meaning of the Copyright Office's examinations useless).

235 *La Resolana Architects*, 416 F.3d at 1202; Kennedy, *supra* note 24, at 341 (stating how "the court was careful to distinguish the actual issuance of a certificate of registration from the official act of registration of the Copyright Office.").

236 *La Resolana Architects*, 416 F.3d at 1202; Prunté v. Universal Music Grp., 484 F. Supp. 2d 32, 38 (D.D.C. 2007); Ward v. Nat'l Geographic Soc., F. Supp. 2d 429, 444-45 (S.D.N.Y. 2002); *see also* Hogan, *supra* note 23, at 861 (explaining "federal standing is not dependent upon the result of an application's view by the Copyright Office" since the court makes its own independent copyright determination, then whether the Copyright Office makes a determination is moot).

237 *La Resolana Architects*, 416 F.3d at 1202. *But see* Kennedy, *supra* note 24, at 341.

238 *La Resolana Architects*, 416 F.3d at 1204; *see* 17 U.S.C. §§ 504, 505 (2010) (providing Copyright statutory damages under section 504 and attorney fees under section 505).

239 *La Resolana Architects*, 416 F.3d at 1207.

In *M.G.B. Homes, Inc. v. Ameron Homes, Inc.*, the plaintiff argued the defendant infringed on its design plan for a home the defendant was constructing.[240] The plaintiff filed its application on May 5, 1986 and its suit on July 3, 1986.[241] The district court stated the plaintiff failed to register its work for initiating a suit; therefore, the court lacked subject-matter jurisdiction.[242] The Eleventh Circuit agreed with the district court in dismissing the plaintiff's claim because the Copyright Office did not make a determination regarding the plaintiff's application.[243] Once the Copyright Office sent the plaintiff the certificate of registration on July 28, 1986, the district court allowed the plaintiff to amend the complaint, which the Eleventh Circuit reasoned was correct, because the district court had subject-matter jurisdiction over the plaintiff's case.[244] The Eleventh Circuit stated, once the plaintiff received the certificate of registration, it could have filed a new lawsuit.[245]

2. Post-*Reed Elsevier, Inc. v. Muchnick*

The Supreme Court in *Reed Elsevier, Inc. v. Muchnick* held a court cannot dismiss a copyright infringement claim based on lack of subject-matter jurisdiction.[246] However, it did not rule on the issue of when registration has occurred under section 411(a) of the Copyright Act.[247] As a result, circuits that choose the registration approach may still dismiss the suit on other grounds.[248]

---

240 M.G.B. Homes, Inc. v. Ameron Homes, Inc., 903 F.2d 1486, 1487 (11th Cir. 1990), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

241 *Id*.

242 *Id*. *But see* Johnson v. Cypress Hill, 641 F.3d 867, 873 (7th Cir. 2011) (stating the plaintiff's assertion that the district court lacked subject-matter jurisdiction was meritless because the Supreme Court had decided subject-matter jurisdiction does not preclude a court from hearing the case).

243 *M.G.B. Homes, Inc.*, 903 F.2d at 1489; *see also* Cain, *supra* note 22, at 385 (arguing "the filed copied does not dictate the scope of an author's infringement action.").

244 *M.G.B. Homes, Inc.*, 903 F.2d at 1489; *see also* Brooks-Ngwenya v. Indianapolis Pub. Schs., 564 F.3d 804, 808 (7th Cir. 2009).

245 *M.G.B. Homes, Inc.*, 903 F.2d at 1489; *see also* Landrigan, *supra* note 215, at 589 (explaining the Tenth Circuit's confusion in La Resolana Architects v. Clay Realtors Angel Fire was because registration means the plaintiff had received certification of registration).

246 Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154, 164–65 (2010) (reasoning jurisdiction refers to a court's statutory or constitutional power to adjudicate a case and not to the parties' rights); *see also* Astle, *supra* note 21, at 478 (discussing Reed Elsevier, Inc. v. Muchnick clarifies how section 411(a) does not state jurisdiction is a requirement for registration, making the copyright infringement claims "more efficient" to process).

247 *Reed Elsevier, Inc*., 559 U.S. at 171; Kennedy, *supra* note 24, at 341.

248 Fourth Estate Pub. Benefit Corp. v. Wall-Street.com, L.L.C., 856 F.3d 1338, 1338–39, 1341 (11th Cir. 2017) (using registration approach and affirming dismissal based on noncompliance with preregistration prior to filing a copyright suit); Kernel Records Oy v. Mosley, 694 F.3d 1294, 1309, 1312 (11th Cir. 2012) (applying the registration approach and affirmed

In *Fourth Estate Public Benefit Corporation v. Wall-Street.com, L.L.C.*, 856 F.3d 1338 (11th Cir. 2017) the plaintiff, who has filed for a petition for certiorari, sent its application to register its articles to the Copyright Office but did not receive approval from the Office or the certificate of registration.[249] The Eleventh Circuit stated the registration application is voluntary, but Congress created incentives to register.[250] Furthermore, the statute requires the Copyright Office to take steps when evaluating the application.[251] The plaintiff has to take action, such as making copies of his or her work, making a deposit with the Copyright Office, and filing the application.[252] Based on the plain language of the Copyright Act, section 410(a) establishes registration after the plaintiff has filed the application and the Copyright Office has examined the application.[253] The Eleventh Circuit held that the plaintiff failed to register its articles before filing an infringement claim and affirmed the lower court's ruling that the plaintiff failed to state a claim.[254]

B. Application Approach

The Fifth and Ninth Circuits have adopted the application approach, which holds once the Copyright Office receives the programmer's application for registration, he or she has fulfilled the requirements of registration.[255]

---

the lower court's grant of summary judgment due to failure to apply for registration prior to initiating a copyright lawsuit); *see* Astle, *supra* note 21, at 463.

249 *Fourth Estate*, 856 F.3d at 1339, *petition for cert. filed*, 86 U.S.L.W. 3330 (U.S. Oct. 13, 2017) (No.17-571).

250 *Id*. *But see* Astle, *supra* note 21, at 488 (explaining how the application approach "promotes the incentives Congress intended" as well as efficiency and fairness).

251 *Fourth Estate*, 856 F.3d at 1340; *see also* 2 NIMMER & NIMMER, *supra* note 22 (explaining how once the applicant files the application materials, he or she has taken all necessary steps, regardless of the Copyright Office's determination).

252 *Fourth Estate*, 856 F.3d at 1341; *see* 17 U.S.C. § 408(a) (2010); 17 U.S.C. § 410(a) (2010); 17 U.S.C. § 410(b) (2010); *see also* Cain, *supra* note 22, at 385 (explaining how, under both approaches, the applicant has to file a copy of the work but the registration approach's argument that Congress intended to require Copyright Office to receive the copy is "unsound.").

253 *Fourth Estate*, 856 F.3d at 1341; *see* 17 U.S.C. § 410(a) (2010); *see also* Landrigan, *supra* note 215, at 599 (explaining how the plain language of the Copyright Act is in accordance with the registration approach). *But see* Foraste v. Brown Uni., 248 F. Supp. 2d 71, 77 (D.R.I. 2003) (registering a work occurs when the Copyright Office receives the application based on the plain meaning of the statute).

254 *Fourth Estate*, 856 F.3d at 1342; *see also* Kernel Records Oy v. Mosley, 694 F.3d 1294, 1311 (11th Cir. 2012) (holding the plaintiff failed to "produce sufficiently probative evidence" that it complied with the statutory prerequisites).

255 Apple Barrel Prods., Inc. v. Beard, 730 F.2d 384, 386–87 (5th Cir. 1984); Cosmetic Ideas, Inc. v. IAC/InteractiveCorp, 606 F.3d 612, 619 (9th Cir. 2010); *see also* Bracey, *supra* note 24, at 133 (stating how the application approach relies on the public policy considerations and the plain language of the Copyright Act)

Under *Apple Barrel Products, Inc. v. Beard*, the plaintiff filed suit over the infringement of a county music program.[256] The plaintiff filed the application but failed to register it.[257] The Fifth Circuit reasoned the plaintiff did not need to possess a certificate of registration.[258] Instead, based on the payment of the required fee and deposit of work, the Fifth Circuit held the receipt of application was sufficient to meet the registration requirement and to allow the suit to move forward.[259] The Fifth Circuit explained ownership of copyrighted material "is shown by proof of originality, copyrightability, and compliance with applicable statutory formalities," which does not involve the registration process.[260]

In *Cosmetic Ideas, Inc. v. IAC/InteractiveCorp*, the plaintiff argued the defendant created and sold a necklace, which was substantially similar to the plaintiff's necklace.[261] The plaintiff asserted it met the registration requirement by receiving confirmation of the application from the Copyright Office before filing a suit.[262] The defendant asserted the Copyright Office had to issue a certificate of registration.[263]

The Ninth Circuit looked at the plain language of section 411(a) of the Copyright Act, but the section provided no guidance for interpreting registration.[264] Then, the Ninth Circuit looked at the whole statute, sections 408 to 412, which discuss copyright registration, and determined the whole statute was ambiguous.[265] The Ninth Circuit noted sections 410(a) and 411(a) require the Copyright Office to take some steps in registration.[266] Since the Copyright Office has the burden of examination and registration, it has a more active role, which implies it has to do more than receive the application.[267] Based on section 411(a), the

---

256 *Apple Barrel Prods.*, 730 F.2d at 386.

257 *Id.* at 386–87.

258 *Id.* at 386; *see* 2 NIMMER & NIMMER, *supra* note 22.

259 *Apple Barrel Prods.*, 730 F.2d at 387; *see also* Hogan, *supra* note 23, at 859 (explaining how the application approach requires fewer formalities, which does not mean this approach eliminates the prerequisite for a lawsuit, and the procedural process is more efficient as a result).

260 *Apple Barrel Prods.*, 730 F.2d at 387; Capitol Audio Access, Inc. v. Umemoto, 980 F. Supp. 2d 1154, 1157 (E.D. Cal. 2013).

261 Cosmetic Ideas, Inc. v. IAC/InteractiveCorp, 606 F.3d 612, 614 (9th Cir. 2010).

262 *Cosmetic Ideas*, 606 F.3d at 614; *see Capitol Audio Access*, 980 F. Supp. 2d at 1157.

263 *Cosmetic Ideas*, 606 F.3d at 614.

264 *Id.* at 616; *see* 17 U.S.C. § 411(a) (2010); *see also* Bracey, *supra* note 24, at 131 (looking at the plain language of the Copyright Act "buttresses" the construction of the act and implies Congress should determine the meaning of registration).

265 *Cosmetic Ideas*, 606 F.3d at 617–18; *see* 17 U.S.C. §§ 408–412 (2010).

266 *Cosmetic Ideas*, 606 F.3d at 617; *see* 17 U.S.C. § 410(a), 411(a); 17 U.S.C. § 411(a) (2010); *see also* Astle, *supra* note 21, at 479, 481 (stating the Congressional intent of section 411(a) was to have "less formality in copyright law," and this section supports the application approach based on the incentives).

267 *Cosmetic Ideas*, 606 F.3d at 617.

Court said Congress intended not only the delivery of the application but also registration process included refusal or acceptance of the application.[268] However, the Court determined section 408(a) is contrary to section 411(a) and chose the application approach because section 408(a) suggests the only requirement to obtain registration is to deliver the application.[269] Nevertheless, the plain language of the statute did not persuade the court.[270] The Ninth Circuit examined the amendments to the Copyright Act including the provision stating registration is optional.[271] Based on the history of the statute, the Court stated that the application view was better in fulfilling congressional intent and providing broad copyright protection.[272]

The Ninth Circuit reasoned the application view was the better approach based on policy considerations.[273] The Court stated, "the application approach avoids unnecessary delay in copyright infringement litigation, which could permit an infringing party to continue to profit from its wrongful acts."[274] This approach allows the Copyright Office to make the evaluation while the parties are litigating the copyright infringement case without causing any prejudice and is aligned with the goal of registration, which is a voluntary process.[275]

Pursuant to section 411(a), the Ninth Circuit held the plaintiff does not have

---

268 *Id.*; *see* 17 U.S.C. § 411(a) (2010); *see also* Iconbazaar, L.L.C. v. Am. Online, Inc., 308 F. Supp. 2d 630, 634 (M.D.N.C. 2004) (interpreting section 410(a) to mean registration does not occur upon filing of the materials but that probably only applies to the issuance of a certification).

269 *Cosmetic Ideas*, 606 F.3d at 617; *see* 17 U.S.C. § 408(a), 411(a) (2010).

270 *Cosmetic Ideas*, 606 F.3d at 617.

271 *Id.*; H.R. REP. NO. 94-1476, at 157 (1976) (registering is a voluntarily process); *see* Astle, *supra* note 21, at 478 (describing section 411(a) as clearly providing an incentive and how this section may be inapplicable if statutory or attorney damages are not to induce him or her).

272 *Cosmetic Ideas*, 606 F.3d at 619; *see also* Kennedy, *supra* note 24, at 343–45 (explaining how the United States Constitution provides copyright protection and balances interests of the owner and the public and Congressional intent to provide broad copyright ownership rights).

273 *Cosmetic Ideas*, 606 F.3d at 619–20; *see* Int'l Kitchen Exhaust Cleaning Assocs. v. Power Washers of N. Am., 81 F. Supp. 2d 70, 72 (D.D.C. 2000) (adopting the application approach "effectuates the interests of justice and promotes judicial economy."); *see also* 2 NIMMER & NIMMER, *supra* note 22.

274 *Cosmetic Ideas*, 606 F.3d at 619; *see* 17 U.S.C. § 412 (2010) (stating a copyright infringement suit must be brought in three years); Bracey, *supra* note 24, at 139 (explaining how if the plaintiff has to wait a couple months to a year for the Copyright Office to make a determination, then he or she may run the statute of limitations); *see also* 2 NIMMER & NIMMER, *supra* note 22, at § 7.16[B][3][b][iii] (stating the issue with backdating the certificate when the statute of limitations has run out).

275 *Cosmetic Ideas*, 606 F.3d at 620-21; Chicago Bd. Of Educ. v. Substance, Inc., 354 F.3d 624, 627, 631 (7th Cir. 2003); Astle, *supra* note 21, at 488; *see also* Hogan, *supra* note 23, at 866 (adopting the registration approach would lead to an imbalance between "creative ownership and cultural process.").

to wait for the Copyright Office to approve or reject the application.[276] In addition, the Court stated once the plaintiff receives the certificate, the registration refers to the data of the application.[277]

V. CONCLUSION

Returning to the programmer who created a program that identifies objects, the programmer wants sufficient copyright protection for his or her program to prevent others from creating a program with similar functions.[278] The programmer wants to enforce his or her legal rights by filing a lawsuit against those who infringe on the program by filing the application with the Copyright Office rather than waiting for a determination or the issuance of the certificate.[279] Congress must update the Copyright Act to provide protection to merged elements of a program and to afford programmers the opportunity to enforce their legal rights against infringers by adopting the application approach.[280]

Some dissenters argue providing more protection to programs would hinder innovation and create monopolies.[281] Granting protection to merged elements of a program would "impede progress in the arts" and would be "contrary to the goals of copyright."[282] Since programs are utilitarian works and courts view programming as an activity involving only a particular result, courts are justified in affording programs with less protection.[283] The purpose of copyright law is to

---

276 *Cosmetic Ideas*, 606 F.3d at 621.

277 *Id.*; *see* Cain, *supra* note 22, at 387 (having a claim for copyright infringement even though the Copyright Office denies validity of copyright).

278 *See* Martyniuk, *supra* note 11; Arenault, *supra* note 9, at 161 (protecting not only the code but also the design of the program); *see also* Conley & Bryan, *supra* note 109, at 581 (1985) (discussing the problems with various substantial similarity tests); Menell, *supra* note 16 (explaining how courts disregard the important features that allow a program to produce a task when they use the merger doctrine); Patry, *supra* note 158, at 53 (protecting the copyright of programs has decreased because the abstraction-filtration-comparison test reduces program from its literal and nonliteral elements to simply its literal elements).

279 *See Cosmetic Ideas*, 606 F.3d at 619–20; Kennedy, *supra* note 24, at 344 (adopting the application approach is correct "in the interests of equity and judicial economy."); Bracey, *supra* note 24.

280 *See* Shaeffer, *supra* note 11, at 410; Astle, *supra* note 21, at 478.

281 Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 46 (D. Mass. 1990); Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 BERKELEY TECH. L.J. 1215, 1292 (2016); *see also* Breyer, *supra* note 33 (explaining how a creator should not receive more protection only because he or she should not be paid less than the social value if the public is to benefit from this creation).

282 Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 838 (10th Cir. 1993); H.R. REP. NO. 60-2222, at 7 (1907).

283 Samuelson, *supra* note 39, at 741; Ambrose, *supra* note 123; *see, e.g.*, Apple Comput., Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1023, 1033, 1047 (N.D. Cal 1992) (reasoning the

promote and reward creative work pursuant to Article I of the Intellectual Property Clause,[284] providing the public to benefit from various creations.[285] Finally, critics contend the policy concerns for expanding the copyright law for programs would undercut the statutory language of section 102(b), which limits the copyright scope of computer programs.[286]

Opponents against more copyright protection for programs fail to understand the nature and design of the program.[287] When courts are determining the copyright protection of a program, they view the program from a user's perspective and do not look at the program as a whole or consider the steps to producing the program, such as the flowchart or the algorithm, which shows the programmer's creativity and skill.[288]

The intent of the copyright law is to balance innovation with public welfare without burdensome requirements.[289] Promoting innovation and maintaining public welfare can occur only if Congress understands the process of creating a program's main function as a whole.[290] To provide programs with copyright protection, Congress should understand each aspect of the program, such as the algorithm and the source code, depend on each other rather than not affording protection based on the merger doctrine.[291] The merger doctrine is an outdated, judicially-created concept that fails to consider the programmer's perspective

---

plaintiff's work purely served a "functional purpose" and, based on the copyright law principles, the work received no protection).

284 U.S. CONST., art. 1, § 8, cl. 8; *see also* United States v. Paramount Pictures, Inc., 334 U.S. 131, 158 (1948) (interpreting the intellectual property clause to reward the author as means to induce him or her to release the work to the public).

285 Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 696 (2d Cir. 1992); Samuelson, *supra* note 281, at 1272.

286 Samuelson, *supra* note 175, at 469; Peter G. Spivack, *Does Form Follow Function? The Idea/ Expression Dichotomy in Copyright Protection of Computer Software in Copyright Protection of Computer Software*, 35 UCLA L. REV. 723, 724 (1988); Teter, *supra* note 172, at 1077; *see* 17 U.S.C. § 102(b) (2010) (stating copyright protection does not extend to "any idea, procedure, process, system, method of operation, concept, principle, or discovery.").

287 Miller, *supra* note 11, at 1047; Mislow, *supra* note 12, at 778, 803; *see also* VERGARI & SHUE, *supra* note 12, at 510–11 (stating how the ever-changing role of technology has outpaced the legal realm).

288 Shaeffer, *supra* note 11, at 341, 345, 348, 368; Dunn, *supra* note 55, at 533 (treating the computer instructions and screen displays together provides the programmer with protection and certainty).

289 Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 620 (7th Cir. 1982), *superseded by statute* FED. R. CIV. P. 52(a) (2017); Michael A. Dryja, *Looking to the Changing Nature of Software for Clues to Its Protection*, 3 U. BALT. INTELL. PROP. L.J. 109, 119–20 (1995).

290 Martyniuk, *supra* note 11, at 1369; Clapes et al., *supra* note 64, at 1545; *see also* Burk, *supra* note 127 (describing programs as blurring the line between the idea and the expression that programs should receive their "own novel brand of intellectual property protection.").

291 McGahn, *supra* note 12; Astle, *supra* note 21, at 467; *see also* 17 U.S.C. § 101 (2010). The source code is the text of the program's instructions and is written in a programming language that resembles English. Ogilvie, *supra* note 61.

and disregards the intent of the copyright law.[292] The purpose of this doctrine is efficiency, and if a programmer uses the most efficient programming techniques to develop an algorithm, the copyright law will not protect the algorithm, and, as a consequence, the programmer is forced to produce an inefficient algorithm.[293]

Some courts assert the registration approach is in alignment with the text and the history of the Copyright Act.[294] On the face of the statute, both the applicant and the Copyright Office have to take action to register the work, and the language does not state the applicant has registered his or her work by mailing the application to the Copyright Office.[295] Instead, the office has to determine the validity of the copyright protection.[296] The language of the Copyright Act also does not provide the programmer with remedies upon filing the application because, to receive those benefits, the programmer has to register his or her work.[297]

Proponents of the registration approach fail to understand the programmer's situation.[298] After the programmer determines the program has sufficient protection, he or she should not have to wait four to five months for the Copyright Office to make a decision on the registration while the infringer continues to make profit.[299] The text of the Copyright Act and the application approach are consistent because the occurrence of registration is separate from the issuance of the certificate of registration.[300] When Congress required registration for lawsuits, the intent was to make it optional.[301] Furthermore, courts do not dismiss

---

292 *See* Mislow, *supra* note 12, at 793 (disqualifying programs for copyright protection is contrary to section 102(b), which intended to differentiate between expressions and ideas); Shaeffer, *supra* note 11, at 346, 368, 410–11; Menell, *supra* note 16, at 1101.

293 Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 708 (2d Cir. 1992); Martyniuk, *supra* note 11, at 1368.

294 *Fourth Estate*, 856 F.3d at 1340-42; *La Resolana Architects*, 416 F.3d at 1200.

295 Kernel Records Oy v. Mosley, 694 F.3d 1294, 1301-02 (11th Cir. 2012); *see* 17 U.S.C. § 410(a) (2010).

296 Gaiman v. McFarlane, 360 F.3d 644, 654-55 (7th Cir. 2004).

297 *Fourth Estate*, 856 F.3d at 1340; Mays & Assocs. v. Euler, 370 F. Supp. 2d 362, 367-70 (D. Md. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010); *see* M.G.B. Homes, Inc. v. Ameron Homes, Inc., 903 F.2d 1486, 1487 (11th Cir. 1990), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

298 *See* Astle, *supra* note 21, at 467; *see, e.g.*, SimplexGrinnell LP v. Integrated Sys. & Power, Inc., 642 F. Supp. 2d 167, 187-88 (S.D.N.Y. 2009) (dismissing infringement suits for alleged copying of plaintiff's other versions because the Copyright Office had not issued a certificate of registration, allowing the defendant to continue to profit off those versions of the program).

299 Hogan, *supra* note 23, at 849–50 (explaining how the purpose of the copyright act was to ease the procedural mechanisms and to protect creators).

300 *Cosmetic Ideas*, 606 F.3d at 617; Iconbazaar, L.L.C. v. Am. Online, Inc., 308 F. Supp. 2d 630, 633-34 (M.D.N.C. 2004); Bracey, *supra* note 24, at 141.

301 Hogan, *supra* note 23, at 846; *see* Foraste v. Brown Uni., 248 F. Supp. 2d 71, 76-77 (D.R.I. 2003) (failing to produce the issuance of certificate for the work is a not "fatal" to the

the requirement of registration for filing a lawsuit; instead, they suggest registration requires fewer formalities.[302]

As a matter of public policy, the application view avoids undue delay in litigation, preventing the infringer from profiting off the programmer's work, and provides equal protection to both plaintiffs who have registered under the registration and application approach.[303] Approval or denial from the Copyright Office is a formality, which will have little impact on the suit because the court determines copyrightability independent of the office's determination.[304] The application view protects the programmer from the infringer having any unfair advantage and advances "the interests of justice" and "judicial economy."[305]

The solution for providing programmers sufficient copyright and litigation protection is two-fold.[306] When the expressions of an idea are incidental to the ideas of a program, those expressions, such as the source code, should not be subject to the merger doctrine.[307] At the end of the section 102(b) of the Copyright Act, Congress should add a line stating, "[f]or purposes of this subsection, computer programs whose ideas and expressions merge are considered protected."[308] This proposed amendment will provide sufficient copyright protection to the expressions of the program and will abrogate the merger doctrine with respect to computer programs.[309] When courts are determining whether the programmer's work is substantially similar to the defendant's program, they would still use the abstraction-filtration-comparison test; however, during the filtration

---

plaintiff's copyright infringement claim).

302 Bracey, *supra* note 24, at 139 (requiring to send both the application and the deposit under both approaches).

303 *Cosmetic Ideas*, 606 F.3d at 619; Kennedy, *supra* note 24, at 344 (adopting the application approach is "in the interest of equity and judicial economy;" therefore, it advances the purpose of copyright law, which is to "progress arts and literature.").

304 Cain, *supra* note 22, at 385; *see, e.g.*, Apple Barrel Prods. v. Beard, 730 F.2d 384, 386-88 (5th Cir. 1984) (determining the copyrightability of the work independent of the Copyright Office's result).

305 Int'l Kitchen Exhaust Cleaning Assocs. v. Power Washers of N. Am., 81 F. Supp. 2d 70, 72 (D.D.C. 2000); Astle, *supra* note 21, at 452; 2 NIMMER & NIMMER, *supra* note 22 (explaining how courts that adopt the registration approach admit their result is inefficient).

306 *See* Maiorana, *supra* note 13, at 188 (advocating for greater copyright protection for the nonliteral aspects of a computer program); Kennedy, *supra* note 24, at 347 (recommending that the courts adopt the application approach).

307 *See* Martyniuk, *supra* note 11, at 1372; Miller, *supra* note 11, at 1041–42.

308 *See* Menell, *supra* note 16; *see also* Green, *supra* note 126 (explaining how Congress would provide a non-inclusive list of nonliteral aspects of a computer program that would be protected); Preonas, *supra* note 147, at 108 (describing how the merger doctrine would not apply to compilations).

309 *See* Faust, *supra* note 127, at 150, 152; *see also* Patry, *supra* note 158, at 42 (explaining the constraints the idea-expression line places on computer programs).

step, the merger doctrine would not apply, which would have removed the protected aspects of the program.[310] Since the copyright law grants "valuable enforceable rights" and incentivizes "individual effort and creativity," programmers will innovate, share their invention, and be confident the copyright law affords protection to those aspects of the program.[311]

This solution does not provide copyright protection to the program's main function or the idea but, instead, to the program's algorithm or source code, which the court would have afforded protection.[312] Furthermore, this solution considers the viewpoint of programmers who have devoted their time and effort to create a program that is aligned with the purpose of the copyright law, to promote progress, because the value of a program lies not only in its function but also in its process.[313]

To ensure courts do not dismiss a lawsuit despite the programmer filing an application for registration before the initiation of a claim for copyright infringement, Congress should adopt the application approach to all works under Section 102(a) of the Copyright Act.[314] Congress will make it easier not only to register a work without having to wait, but also place others on notice.[315] Congress would clarify Section 411(a) of the Copyright Act by stating, "[f]or purposes of this subsection, registration will be deemed to have occurred when the deposit, application, and fee required for registration have been received by the Copyright Office in proper form."[316] With the adoption of the application approach, courts will be more consistent, effective, and reasonable because programmers will not have to worry about the infringer making profit from his or her work and the

---

310 *See* Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706-10 (2d Cir. 1992); Mark E. Dailey, *supra* note 195, at 434–35.

311 Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp., 672 F2d 607, 620 (7th Cir. 1982), *superseded by statute* FED. R. CIV. P. 52(a) (2017); *see* S. REP. NO. 94-473, at 51 (1975); *Copyright Protection of Computer Program Object Code*, *supra* note 65, at 1739.

312 *See* 17 U.S.C. § 102(b); Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 (3d Cir. 1986) (defining the idea-expression line); Velasco, *supra* note 5, at 279.

313 *See* Shaeffer, *supra* note 11, at 346, 368; Arenault, *supra* note 9, at 160–61 (explaining how the programmer has an interest in protecting the "ideas within the system," not the "idea of the system."); *see also* Mislow, *supra* note 12, at 803 (arguing Congress should recognize the program is "valuable" and the need to specify the important aspects of a program, such as the microcode, require protection).

314 17 U.S.C. § 102(a) (2010); *see* Cain, *supra* note 22, at 385, 391; *see also* Olander Enterprises, Inc. v. Spencer Gifts, LLC., 2011 WL 13225062 at *3 (C.D. Cal. Oct. 21, 2011) (acknowledging that submitting an application creates standing for a copyright claim but does not subrogate a registration as validation).

315 *See* Bracey, *supra* note 24, at 139.

316 Astle, *supra* note 21; *see* Kennedy, *supra* note 24 (arguing courts should apply application approach because this approach facilitates "the progress of arts and sciences.").

public will have access to a national registry.[317]

317 *See Cosmetic Ideas*, 606 F.3d at 619-20; Hogan, *supra* note 23, at 853; *see also* 2 NIMMER & NIMMER, *supra* note 22, at § 7.16[B][3][b][vi] (illustrating an example of how the registration approach is "untenable.").