

2018

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

Akshay Jain

Catholic University of America (Student)

Follow this and additional works at: <https://scholarship.law.edu/jlt>



Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), [Intellectual Property Law Commons](#), and the [Science and Technology Law Commons](#)

Recommended Citation

Akshay Jain, *Providing Protection to Programmers' Works: Disregard the Merger Doctrine and Adopt the Application Approach*, 26 Cath. U. J. L. & Tech 129 (2018).

Available at: <https://scholarship.law.edu/jlt/vol26/iss2/6>

This Comment is brought to you for free and open access by CUA Law Scholarship Repository. It has been accepted for inclusion in Catholic University Journal of Law and Technology by an authorized editor of CUA Law Scholarship Repository. For more information, please contact edinger@law.edu.

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 similar functions.⁴ The programmer files a complaint against the

⁺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).

⁴ 17 U.S.C. § 106 (2012) (stating "[t]he owner of copyright under this title has 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. Applying the merger doctrine defeats the purpose of creating a program and the Congressional intent of the copyright law because the programmer created a program to distribute to the public and to promote innovation.⁹ The copyright law “encourage[s] individual effort and creativity by granting valuable enforceable rights.”¹⁰

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).

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).

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) (dismissing 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).

⁹ 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).

¹⁰ *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 also* 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 (1993) (explaining how technology is rapidly changing and, as a result, is challenging the scope of copyright law).

Congress must view the copyright scope of a program from the programmer's perspective, understanding the artistic process to create a program. Furthermore, computer programs differ from other types of literary works because the nature and design of programs are highly complex.¹¹ Congress must amend the Copyright Act by abrogating the judicially created concept, the merger doctrine, and not apply it to programs.¹² 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.

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.¹³ To sue for copyright infringement, the programmer has to register his or her program.¹⁴ However, when the Court considers the work was registered, the approaches among the circuits are split.¹⁵ Courts adopt either the registration¹⁶ or application¹⁷ approach.¹⁸

¹¹ 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); John Shaeffer, *Software as Text*, 33 SANTA CLARA COMPUT. & HIGH TECH. L.J. 324, 410 (2017) (comparing and contrasting different types of literary works with the complexity of programs); see also JAMES V. VERGARI & VIRGINIA V. SHUE, FUNDAMENTALS OF COMPUTER-HIGH TECHNOLOGY LAW, 1 (American Law Institute-American Bar Association Committee on Continuing Professional Education 1991) (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 protection literary works receive).

¹² The amendment to the Copyright Act would include other types of computer programs.

¹³ See 17 U.S.C. § 411 (2010) (describing the procedure to file a lawsuit for copyright infringement).

¹⁴ 17 U.S.C. § 411(a) (2010).

¹⁵ 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).

¹⁶ 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 a claim for copyright infringement).

¹⁷ 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 the registration of the work occurred because the Copyright Office received the plaintiff's application).

¹⁸ *La Resolana Architects v. Clay Realtors Angle Fire*, 416 F.3d 1195, 1202 (10th Cir.

Congress must adopt the application approach because programmers should 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. Based on the language and history of the Copyright Act, registration requires fewer formalities.¹⁹ Under the application approach, Courts would become more efficient and predictable, providing programmers with the opportunity to enforce their legal rights.²⁰

Part I of this Comment will provide background on application programs and describe the process to create an application program. Part II will discuss what is copyright and then will begin with the history of the copyright law with respect to computer programs. 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. Part III puts forth the elements required to prove copyright infringement. 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. Part IV of this Comment explains the circuit split over the registration requirements for litigation purposes. 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. Consequently, programmers will have an expectation of the type of protection for their programs.

I. BACKGROUND ON APPLICATION PROGRAM

An average person's colloquial use "application program" is often

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).

¹⁹ *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); 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).

²⁰ 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).

interchangeable with “computer program.”²¹ In this Comment, “application program” and “computer program” have different meanings. An “application program” is a program that allows “a user to perform some particular task such as word processing.”²² 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 algorithm.”²³ The programmer can divide the computer program into the operating system program²⁴ and the application program.²⁵

²¹ 1–2 DAVID BENDER, *COMPUTER LAW* § 2.06, 2 (2017), Lexis (database updated Dec. 2017). 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 11, 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 interacting with the software, which is the computer program, and the hardware, which is the computer).

²² *Lotus Dev. Corp. v. Paperback Software Int’l*, 740 F. Supp. 37, 43 (D. Mass. 1990); *see also* Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 *STAN. L. REV.* 1045, 1051 (1989) (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).

²³ VERGARI & SHUE, *supra* note 11, at 508. A computer program can also be defined as a “list of instructions that the computer executes to achieve the result desired by the programmer.”; Velasco, *supra* note 5, at 244. However, the Copyright Act 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”; 17 U.S.C. § 101(a) (2010). An algorithm is a “step-by-step formula for accomplishing a task.” SCHNEIDER ET AL., *supra* note 2, at 3.

²⁴ The discussion of this paper will focus on application programs. 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 11, 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 21, at § 2.06, 2–3.

²⁵ Howard Root, *Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test*, 68 *MINN. L. REV.* 1264, 1268 (1984). An application program is “[a] computer software program designed for a specific job.” *Application Program*, NEWTON’S TELECOMM DICTIONARY (28th ed. 2014). 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. Menell, *supra* note 22, at 1051.

The purpose of an application program (“program”) is to solve the particular problem.²⁶ 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.²⁷

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.²⁸

1. Defining the Problem

The first step is to determine “what purpose the [program] will serve for the user.”²⁹ The programmer has to analyze the problem by gathering and examining information required to produce the aspired result.³⁰ 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.”³¹ Once the

²⁶ 1–2 BENDER, *supra* note 21 (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).

²⁷ 1–2 BENDER, *supra* note 21; *see also* J.M. YOHE, AN OVERVIEW OF PROGRAMMING PRACTICES 224 (Association for Computing Machinery, Inc. 1974) (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).

²⁸ YOHE, *supra* note 27, at 222. Computer programming is “the entire series of steps involved in solving a problem on a computer.” SCHNEIDER ET AL., *supra* note 2, at 1. 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).

²⁹ 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 11, 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).

³⁰ 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.

³¹ SCHNEIDER ET AL., *supra* note 2, at 7.

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 data³² specifications.³³

2. Designing the Structure

Once the programmer decides the program’s purpose, he or she defines the purpose in a series of steps.³⁴ The programmer will describe those steps by writing a pseudo code³⁵ or creating a flowchart.³⁶ To convey that representation, the programmer has to design the structure by creating data structures³⁷ and algorithms.³⁸ Data structures are variables that hold data, and an algorithm takes the data structure, analyzes it, and has the computer execute it.³⁹ An algorithm

³² The input data refers to the format and the output data refers to specifications. 1–2 BENDER, *supra* note 21, at § 2.06 [3][b].

³³ YOHE, *supra* note 27, 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).

³⁴ VAZSONYI, *supra* note 21, at 291; *see also* YOHE, *supra* note 27, 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.”).

³⁵ 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), Lexis (database updated Dec. 2017); *see also* LAWRENCE J. PETERS, *SOFTWARE DESIGN: METHODS & TECHNIQUES* 93 (Yourdon Press, 1981) (describing how a visual of pseudo code does not consist of syntax or semantics).

³⁶ John Swinson, *Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection*, 5 HARV. J.L. & TECH. 145, 148 (1991). 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 10, at 1342; *see also* VAZSONYI, *supra* note 21, at 151 (providing an illustration of a flowchart).

³⁷ 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). 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 27, at 226.

³⁸ SCHNEIDER ET AL., *supra* note 2, at 3, 19. 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 11, at 24.

³⁹ 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

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.⁴⁰ The programmer has to decompose the algorithm into subroutines.⁴¹ 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.⁴² The programmer produces a flowchart or a pseudo code to assist the programmer with designing and visualizing the program.⁴³

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.”⁴⁴ This process is known as coding.⁴⁵

requirements. See Michael S. Gal & Niva Elkin-Koren, *Algorithmic Consumers*, 30 HARV. J.L. & TECH. 309, 313–14 (2017).

⁴⁰ YOHE, *supra* note 27, 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 pseudo code or a flowchart, he or she describes the program at a high level “with no particular processor in mind.” Swinson, *supra* note 36, at 148.

⁴¹ 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); 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, 147 (1994) (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. MIGHTOM, *A STRUCTURED APPROACH TO PROGRAMMING* 21, 23 (Prentice-Hall, Inc. 1977).

⁴² LEEDS & WEINBERG, *supra* note 30, at 212; Velasco, *supra* note 5, at 246.

⁴³ VAZSONYI, *supra* note 21, at 292; SCHNEIDER ET AL., *supra* note 2, at 34.

⁴⁴ VERGARI & SHUE, *supra* note 11, at 24–25; Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281, 341–42 (1970) (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.”).

⁴⁵ YOHE, *supra* note 27, at 231; see also VAZSONYI, *supra* note 21, 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.

First, the programmer implements⁴⁶ the pseudo code or flowchart by writing each step of the algorithm in a programming language.⁴⁷ This is because the computer cannot execute⁴⁸ a human language since it does not understand it.⁴⁹ The type of programming language a programmer chooses reveals the level of training as well as the style and creativity.⁵⁰ A programmer can write the program in different programming languages, such as machine,⁵¹ assembly,⁵² and high-level.⁵³

⁴⁶ The programmer takes the design of the program in flowchart form and implements it into machine-readable code. Menell, *supra* note 22, 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 FORTRAN.”).

⁴⁷ A programming language is “a set of symbols and syntax rules used to express strings of bits.” Holton, *supra* note 21, 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. INTEL. PROP. L. 1, 16 (2016). A programming language can be defined as a 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).

⁴⁸ 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).

⁴⁹ Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230 (3d Cir. 1986); VAZSONYI, *supra* note 21, at 103; *see also* Dunn, *supra* note 41, at 501 (explaining how computers operate on a machine language and not on a human language).

⁵⁰ 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 et al., *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).

⁵¹ 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).

⁵² 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).

⁵³ Hamilton & Sabety, *supra* note 37, 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). Programmers use high-level languages, such as FORTRAN (Formula Translation) for mathematical statements, or COBAL (Common Business Oriented Language) for business data processing. Root, *supra* note 25, at 1266 n.12.

However, most programmers code⁵⁴ in a language that is readable by humans.⁵⁵ Then, he or she uses a compiler⁵⁶ or an interpreter,⁵⁷ depending on the programmer's needs,⁵⁸ to convert the program in a source code⁵⁹ into an object code,⁶⁰ which computers can read.⁶¹ Once the programmer has produced an object code, the computer can execute the written instructions and produce the desired result.⁶²

i. Machine

A machine language is “a programming language composed of machine instructions that can be executed directly by a computer without further

⁵⁴ 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).

⁵⁵ 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); Steven R. Englund, *Ideas, Process, or Protected Expression: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 868 (1990) (stating how “computers are only capable of directly executing ‘computer instructions’”).

⁵⁶ 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). A compiler is also defined as translating a high-level language into a machine code. VERGARI & SHUE, *supra* note 11, at 26.

⁵⁷ An interpreter translates one statement at a time from source code to object code.

Christopher M. Mislow, *Computer Microcode: Testing the Limits of Software Copyrightability*, 65 B.U.L. REV. 733, 744 n.45 (1985).

⁵⁸ 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. *Id.* at 744 n.45. A CPU, or computer-processing unit, is the “essential core of the computer system” and are the computer’s “brains.” VERGARI & SHUE, *supra* note 11, 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).

⁵⁹ 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.”).

⁶⁰ 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 41, at 501 (explaining how an object code as “either [an] executable machine code or code that can be processed easily into machine code.”).

⁶¹ VAZSONYI, *supra* note 21, at 103.

⁶² Ogilvie, *supra* note 47, at 531; Samuelson, *supra* note 26, at 686.

compilation” and is the only language recognized by a machine.⁶³ It represents the instructions in binary form, consisting of zeros and ones.⁶⁴ For example, the instruction 010001 could mean run program.⁶⁵

Programmers usually do not write their algorithms in machine language because it is difficult for them to understand and it is tedious.⁶⁶ Instead, a programmer writes the program in a readable language and produces a source program.⁶⁷ Then, he or she converts the source program into an object program,⁶⁸ which consists of strings of binary digits.⁶⁹ Upon conversion, the computer reads the language, executes the algorithm, and produces the expected result.⁷⁰

⁶³ *Machine language*, NEWTON’S TELECOMM DICTIONARY (28th ed. 2014); Peter D. Aufrichtig, *Copyright Protection for Computer Programs in Read Only Memory Chips*, 32 COPYRIGHT L. SYMP. 133, 146 (1982) (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 $C = A + B$.”).

⁶⁴ Clapes et al., *supra* note 50, 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).

⁶⁵ Mislow, *supra* note 57, at 743; *see also*, Davidson, *supra* note 29, at 616 (representing “100110010” as an instruction to add “register B to whatever is in the accumulator.”).

⁶⁶ *Copyright Protection of Computer Program Object Code*, *supra* note 51, at 1725; *see also* Lotus, 740 F. Supp. at 44 (explaining how programmers rarely write directly on the computer); YOHE, *supra* note 27, at 236 (describing that choosing a programming language is based on one’s experience and natural familiarity).

⁶⁷ VERGARI & SHUE, *supra* note 11, 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).

⁶⁸ 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.”).

⁶⁹ Reger, *supra* note 59, at 219; Root, *supra* note 25, 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 21, at 196.

⁷⁰ Samuelson, *supra* note 26, at 686; *see also* VAZSONYI, *supra* note 21, 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 result).

ii. Assembly

An alternative to machine language is the assembly language for writing his or her program.⁷¹ Assembly language is represented in mnemonic or alphanumeric instructions.⁷² For example, the instruction “ADD X” could mean Add lane X to the register.⁷³ 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.⁷⁴

iii. High Order

The most popular and widely used programming language is the high-level language because it uses “English-like”⁷⁵ words and is easier to understand and program⁷⁶ than the machine or assembly language.⁷⁷ 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.”⁷⁸ The programmer translates the algorithms that are components of the program in a high-level language into

⁷¹ Shaeffer, *supra* note 11, at 351 (stating each “assembly language instruction corresponds to a single machine-language instruction.”); Dunn, *supra* note 41 (explaining how “[humans] can read assembly language” but it resembles closer machine language).

⁷² *Copyright Protection of Computer Program Object Code*, *supra* note 51, at 1725; Aufrechtig, *supra* note 63, 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).

⁷³ Davidson, *supra* note 29, at 620.

⁷⁴ Root, *supra* note 25, at 1267; VAZSONYI, *supra* note 21, 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).

⁷⁵ Davidson, *supra* note 29, 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).

⁷⁶ Program means “[t]o design, write, and test computer programs.” *Program*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Ogilvie, *supra* note 47, at 530 (defining a program, or a software that runs on the computer, as “an organized set of instructions that guides a computer.”).

⁷⁷ Dunn, *supra* note 41, at 500 (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).

⁷⁸ *High-level language*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984); *see also* Menell, *supra* note 22, at 1056 (stating while there are various types of high-level languages, what matters is the “accuracy, efficiency, and reliability of the resulting program.”).

a machine code because the computer “can only deal with machine language programs written in bit strings.”⁷⁹ Once the translator⁸⁰ converts the source program into an object program, the program is machine readable.⁸¹ Then, the programmer can run the program.⁸²

4. *Debugging and Testing*

Before the programmer begins to debug⁸³ and test⁸⁴ 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.⁸⁵

Once the programmer reviews and proofreads the program, he or she runs the program to determine the errors⁸⁶ or bugs⁸⁷ in the program.⁸⁸ Debugging consists of find and correcting all errors causing the program to produce “either incorrect results or no results.”⁸⁹ Testing is a process “in which a program is validated”

⁷⁹ Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230-31 (3d Cir. 1986); VAZSONYI, *supra* note 21, at 103; Swinson, *supra* note 36, at 149.

⁸⁰ 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 compiler).

⁸¹ Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 838 (Fed. Cir. 1992); Dunn, *supra* note 41, at 501.

⁸² 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).

⁸³ 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 22, at 1059 (debugging is a process that involves “testing the program for accuracy, correcting programming errors, and verifying that the program functions properly.”).

⁸⁴ Testing means comparing the produced results to the expected results. GILBERT, *supra* note 33, at 13.

⁸⁵ YOHE, *supra* note 27, at 236, 239–40.

⁸⁶ 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 (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.”).

⁸⁷ 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 30, at 66.

⁸⁸ YOHE, *supra* note 27, at 236 (stating the programmer can conduct the compilation process before testing).

⁸⁹ SCHNEIDER ET AL., *supra* note 2, at 204. The debugging procedure has two phases,

and “the how and what to test can be specified.”⁹⁰ When the programmer is debugging and testing, he or she is concerned with finding and correcting all possible errors.⁹¹ 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.⁹² The programmer continues this process until all known errors are removed.⁹³ This stage is probably the most frustrating and time-consuming task to the programmer.⁹⁴

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.⁹⁵ When the program fails to run as expected, the programmer has to restart the debugging and testing process.⁹⁶ After each run, the programmer should thoroughly examine the results and determine what corrections must be made.⁹⁷

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 21, at § 2.06 [3][f].

⁹⁰ HUGHES & MIGHTOM, *supra* note 41, at 237 (stating 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> (“testing as a user and as a developer are white box and black box testing” 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.”).

⁹¹ Samuelson, *supra* note 26, at 687; *see also* VAZSONYI, *supra* note 21, at 293 (explaining how debugging and testing is about finding errors and achieving the intended result).

⁹² HUGHES & MIGHTOM, *supra* note 41, at 247; YOHE, *supra* note 27, at 238.

⁹³ 1–2 BENDER, *supra* note 21, at § 2.06 [3][f]; *see* VERGARI & SHUE, *supra* note 11, 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).

⁹⁴ 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).

⁹⁵ HUGHES & MIGHTOM, *supra* note 42, 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).

⁹⁶ HUGHES & MIGHTOM, *supra* note 41, at 237.

⁹⁷ YOHE, *supra* note 27, at 238; *see also* 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

If the programmer believes the program is running properly and is producing the expected result, then he or she can verify the program.⁹⁸ The programmer continues this process of finding all conceivable errors even after dissemination of the program.⁹⁹

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.¹⁰⁰ During documentation, the programmer is “creating, collecting, organizing, storing, and communicating information necessary to use the program.”¹⁰¹ The purpose of documentation is to communicate the program to other programmers who are interested in modifying the program.¹⁰² 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.¹⁰³

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.¹⁰⁴ Having good documentation will assist the programmer with

assure all defects have, in fact, been removed.”).

⁹⁸ YOHE, *supra* note 27, at 239 (verifying can be done by running the program on a different computer or an independent calculation).

⁹⁹ HUGHES & MICHOM, *supra* note 41, at 237; 1–2 BENDER, *supra* note 21, at § 2.06 [3][f] (explaining how coding changes are required to remove the bugs and must be careful to not introduce new bugs).

¹⁰⁰ 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 27, at 240.

¹⁰¹ VAZSONYI, *supra* note 21, 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).

¹⁰² 1–2 BENDER, *supra* note 21, at § 2.06 [3][g]; *see also* Menell, *supra* note 22, at 1059 (documenting entails “preparing materials that explain the functioning of the program.”).

¹⁰³ YOHE, *supra* note 27, 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.”).

¹⁰⁴ YOHE, *supra* note 27, 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

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.¹⁰⁵ After the programmer makes the necessary changes, he or she should update the documentation to reflect any bug fixes and enhancements per new features.¹⁰⁶

Once the program is functional, the user does not see the code or the design and is concerned with utility and functionality of the program.¹⁰⁷ Many commentators and courts view programming as an activity that involves achieving a particular result, thereby ignoring the process of creating a program.¹⁰⁸ 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.¹⁰⁹ Essentially, the creation of a program is not "a scientifically objective process."¹¹⁰ Instead, courts must understand the programmer put considerable time and creativity into creating the program by developing flowcharts or pseudo codes and designing algorithms.¹¹¹

required function." *Maintenance*, AMERICAN NATIONAL DICTIONARY FOR INFORMATION PROCESSING SYSTEMS (1984).

¹⁰⁵ YOHE, *supra* note 27, at 240–41; *see also* PIÑEIRO, *supra* note 50, at 127 (explaining how having a readable code makes it easier for the programmer to maintain).

¹⁰⁶ SCHNEIDER ET AL., *supra* note 2, at 5.

¹⁰⁷ David G. Luetgen, *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 26, 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).

¹⁰⁸ Reger, *supra* note 59, 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).

¹⁰⁹ VERGARI & SHUE, *supra* note 11, at 509; Arenault, *supra* note 42, 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.").

¹¹⁰ Shaeffer, *supra* note 11, at 346; *see also* PIÑEIRO, *supra* note 51, at 242–43 (explaining how software is not an objective piece of mathematical code but a subjective design that accomplishes a goal by one of many possible methods); Clapes et al., *supra* note 50, 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).

¹¹¹ Mislow, *supra* note 57, 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. & ENTMT. L.J. 89, 135 (1992); *see also* Menell, *supra* note 22, at 1053–54 (explaining how the human factor assists in developing and expanding the design and the coding of application programs).

Consequently, 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.¹¹²

After the programmer is confident about his or her application program, the programmer can choose to disseminate it to the public.¹¹³ When contemplating this decision, the programmer may have concerns about the unlawful use of his or her work.¹¹⁴ Therefore, the programmer should obtain copyright protection by registering his or her work.¹¹⁵ 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.¹¹⁶

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.”¹¹⁷ 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.¹¹⁸ In 1988, Congress amended the Copyright Act of 1976 and added

¹¹² 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. 1983).

¹¹³ YOHE, *supra* note 27, at 240–41; *see* Luetzgen, *supra* note 107 (stating how programs exist to be functional, such as WordPerfect function for print, which prints pages when directed).

¹¹⁴ Velasco, *supra* note 5, at 248 (explaining how the user interface may not be protected).

¹¹⁵ 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) (explaining how the plaintiff registered its video game because it believed the defendant copied the audiovisuals and the machinations of the video characters).

¹¹⁶ 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 115.

¹¹⁷ U.S. CONST., art. 1, § 8, cl. 8.

¹¹⁸ 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

computer programs as a literary work.¹¹⁹ Notably, copyright is not defined in Section 101 of the Copyright Act.¹²⁰

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.¹²¹ 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.¹²² 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."¹²³ While programmers may have copyright protection for their

forms of protection).

¹¹⁹ 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 also* S. REP. NO. 94-473, at 50 (1975) (explaining how Congress intended to protect computer programs based on the history of the copyright 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."

¹²⁰ 17 U.S.C. § 101 (2010).

¹²¹ *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 119, at 12; *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).

¹²² 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 119, at 63. "Literary 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." 17 U.S.C. § 101; *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).

¹²³ 17 U.S.C. § 101 (2010).

programs, the statute provides limited protection.¹²⁴

B. The Requirements of Copyright Protection

For a program to have copyright protection, it must meet the elements of originality¹²⁵ and fixation.¹²⁶ However, the plain language of the statute does not state those requirements.¹²⁷ Nonetheless, courts have determined originality and fixation as constitutional and statutory requirements for copyright protection.¹²⁸ The purpose of copyright law was to protect the “original expression of any creative effort and not the effort itself.”¹²⁹

1. Originality

For a work to have originality, “the work owes its origin to the author.”¹³⁰ Originality requires the creator to “independently create” and to “possesses at least some minimal degree of creativity.”¹³¹ While the subsequent work may be

¹²⁴ Martyniuk, *supra* note 10, at 1340; *see also* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 833 (10th Cir. 1993) (stating that “[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.”); 17 U.S.C. § 102(a) (2010) (providing computer software is considered copyrightable material).

¹²⁵ Originality means “an original work of authorship in the expression of ideas.” VERGARI & SHUE, *supra* note 11, 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).

¹²⁶ 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 41, 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 11, 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).

¹²⁷ 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).

¹²⁸ Feist Publ’ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 351, 355 (1991).

¹²⁹ 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).

¹³⁰ 1 NIMMER & NIMMER, *supra* note 35, at § 2.01[A][1].

¹³¹ 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.”); L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486, 490

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.¹³² For instance, if two programmers create the same program unbeknownst to them, both programs are copyrightable because each program was independently created. The required level of creativity is “extremely low” that “a slight amount will suffice.”¹³³

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.”¹³⁴ For example, a live performance of a song or an image is not fixed, but a program is fixed. The difference between those works is that a program is in tangible form and is “sufficiently permanent” or can be reproduced.¹³⁵ Usually, plaintiffs can prove originality and fixation by showing to the court he or she registered the work with the Copyright Office.¹³⁶

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.¹³⁷ Under

(2d Cir. 1976) (explaining originality does not require novelty or ingeniousness).

¹³² *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 little creativity to qualify for originality).

¹³³ *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).

¹³⁴ 17 U.S.C. § 101 (2010); see also 1 NIMMER & NIMMER, *supra* note 35, 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).

¹³⁵ 17 U.S.C. § 101 (2010); 1 NIMMER & NIMMER, *supra* note 35, at § 2.03[B][2]; 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).

¹³⁶ 17 U.S.C. § 408(a) (2010); 4 NIMMER & NIMMER, *supra* note 35, at § 13.01[A]; see also *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485, 1492 (D. Minn. 1985) (holding the plaintiff met the requirement of originality because it received the certification of registration from the Copyright Office for the ROM).

¹³⁷ *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 834 (10th Cir. 1993); see

the Copyright Act, a programmer's work does not have protection to his or her ideas, but rather to the expression of those ideas.¹³⁸ In other words, copyright protection does not extend to the main function of the work.¹³⁹ As a consequence, courts rely on the merger doctrine and employ other tests to determine what parts of the program are protected.¹⁴⁰

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.¹⁴¹ While an individual could not copy the programmer's code, one could produce the same end result or namely, the idea itself.¹⁴² All programs have literal and non-literal aspects.¹⁴³ The literal aspects of a program are the source, object code, and the flowchart, and the non-literal aspects are the structure, sequence, organization, and user interface.¹⁴⁴ While literal parts of a program are protected if original,¹⁴⁵ non-literal parts may be protected or unprotected because the non-literal aspects are functional in nature and thus can

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).

¹³⁸ 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).

¹³⁹ *Gates Rubber Co.*, 9 F.3d at 833; *see also* Martyniuk, *supra* note 10, at 1343 (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).

¹⁴⁰ *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 11, 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.

¹⁴¹ *Comput. Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 703 (2d Cir. 1992). The source code is “[t]he literal text comprising the program's instructions.” *Ogilvie*, *supra* note 47. 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).

¹⁴² *Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1993); *Conley & Bryan*, *supra* note 95, at 571.

¹⁴³ *Comput. Assocs. Int'l*, 982 F.2d at 701; Jeffrey Malkan, *Rule-Based Expression in Copyright Law*, 57 *BUFF. L. REV.* 433, 451 (2009).

¹⁴⁴ *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 35, at § 2A.10[B] (defining the object code as a “machine-readable language.”); *see also* *Menell*, *supra* note 22, at 1048 (explaining how various courts have held the literal parts of a program are the source and object code).

¹⁴⁵ *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).

be unprotected.¹⁴⁶ Establishing which part of a program is an idea or an expression is an issue attorneys frequently litigate.¹⁴⁷

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.”¹⁴⁸ 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 and the algorithm, is protected.¹⁴⁹ 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.¹⁵⁰ 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.¹⁵¹ Furthermore, courts have heavily criticized this test and instead apply

¹⁴⁶ *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) (“defining 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).

¹⁴⁷ *Comput. Assocs. Int’l*, 982 F.2d at 705; see also Maiorana, *supra* note 11, 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).

¹⁴⁸ *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).

¹⁴⁹ See *Whelan Assocs.*, 797 F.2d at 1236; see also Clapes et al., *supra* note 50, at 1552.

¹⁵⁰ *Whelan Assocs.*, 797 F.2d at 1248; Reger, *supra* note 59, at 225.

¹⁵¹ Shaeffer, *supra* note 11, at 346, 368; *Paycom Payroll, LLC*, 758 F.3d at 1205; see

other tests to determine what is the original expression and what parts are protectable and unprotectable of a program.¹⁵²

2. *The Merger Doctrine*

When elements of a program include both the idea and the expression of the idea, courts apply the merger doctrine.¹⁵³ 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.”¹⁵⁴ When the defendant admits to copying some portions of the programmer’s work, the defendant can use 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.¹⁵⁵ 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.¹⁵⁶

Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (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 that courts consider a flowchart to be protected, “if sufficiently detailed and original.”).

¹⁵² 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).

¹⁵³ *Comput. Assocs. Int’l*, 982 F.2d at 708; *see also* Englund, *supra* note 55, 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).

¹⁵⁴ *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 protect 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, supra* note 111.

¹⁵⁵ *ATC Distrib. Grp., Inc. v. Whatever It Takes Transmission & Parts, Inc.*, 402 F.3d 700, 709 (6th Cir. 2005); *see also* 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).

¹⁵⁶ *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).

When a court makes inquiries 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.¹⁵⁷ Courts apply the merger doctrine frequently to copyright infringement claims because they want to incentivize competition and promote efficiency.¹⁵⁸ 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 of experience and knowledge to produce an intelligent and functioning program.¹⁵⁹ Consequently, Congress should not deny a programmer's work copyright protection.¹⁶⁰

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.¹⁶¹ 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."¹⁶²

A. Ownership of a Valid Copyright

The programmer can usually demonstrate "ownership of a valid copyright"

¹⁵⁷ Shaeffer, *supra* note 11, at 368; *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 838 (10th Cir. 1993).

¹⁵⁸ *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).

¹⁵⁹ Shaeffer, *supra* note 11, at 370; *see also* Faust, *supra* note 112 (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 22, at 1101 (describing how programming is a human learning process and courts should be cognizant of the important features that the programmer creates).

¹⁶⁰ Shaeffer, *supra* note 11, at 370.

¹⁶¹ 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).

¹⁶² 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).

by presenting to the court the certificate of registration.¹⁶³ 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.”¹⁶⁴ 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.¹⁶⁵

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.¹⁶⁶ First, the court conducts a factual inquiry as to whether the defendant copied aspects of the programmer’s program.¹⁶⁷ 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.¹⁶⁸

1. Factual Copying

To determine factual copying, the programmer must show the defendant had the opportunity to view or access the program.¹⁶⁹ The programmer can provide direct proof to the court.¹⁷⁰ However, proving the defendant had physical access to the programmer’s work is difficult.¹⁷¹ Instead, the programmer can

¹⁶³ *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.

¹⁶⁴ *Gates Rubber Co.*, 9 F.3d at 831–32; *see also* Dunn, *supra* note 41, at 500 (finding that copyrighting is more complicated than originally thought).

¹⁶⁵ *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 that since the plaintiff had the certificates of registration, which constituted the “prima facie evidence of the validity of the plaintiff’s copyright.”).

¹⁶⁶ *Feist Publ’ns, Inc.*, 499 U.S. at 361–62; *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).

¹⁶⁷ *Gates Rubber Co.*, 9 F.3d at 832.

¹⁶⁸ *Arnstein v. Porter*, 154 F.2d 464, 468 (2d Cir. 1946).

¹⁶⁹ *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 838 (Fed. Cir. 1992); *see also* 4 NIMMER & NIMMER, *supra* note 35, at § 13.01[B] (describing the inquiry as factual and is not concerned with the legal question of whether the copying was substantial).

¹⁷⁰ *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); *VERGARI & SHUE, supra* note 11, 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.”).

¹⁷¹ *Johnson v. Gordon*, 409 F.3d 12, 18 (1st Cir. 2005); Dunn, *supra* note 41, at 509;

demonstrate access through indirect proof by arguing the defendant had a “reasonable opportunity” to appropriate the programmer’s work.¹⁷²

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.¹⁷³ 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.¹⁷⁴ 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.¹⁷⁵ As a result, courts have developed various tests to determine the extent to which the programmer’s work has copyright protection.¹⁷⁶

Olufunmilayo B. Arewa, *The Freedom to Copy: Copyright, Creation, and Context*, 41 U.C. DAVIS 477, 486 (2007).

¹⁷² 4 NIMMER & NIMMER, *supra* note 35, 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. v. Bando Chem. Indus. Ltd.*, 9 F.3d 823, 833 n.8 (10th Cir. 1983).

¹⁷³ *Feist Publ’ns, Inc. v. Rural Tel. Servs. Co., Inc.*, 499 U.S. 340, 361 (1991).

¹⁷⁴ *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).

¹⁷⁵ *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1355 (Fed. Cir. 2014); *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989); *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 11, at 554–55.

¹⁷⁶ *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); Donald F. McGahn II, *Copyright Infringement of Protection Computer Software: An Analytical Method to Determine Substantial Similarity*, 21 RUTGERS COMPUTER & TECH. L.J. 88, 113 (1995) (employing tests, such as the ordinary observer, the extrinsic/intrinsic, and the total concept and feel, to determine the copyright protection of a program).

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.¹⁷⁷ 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.¹⁷⁸

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.¹⁷⁹ The court divides the program into six levels: the main function,¹⁸⁰ the source code,¹⁸¹ the object code,¹⁸² modules,¹⁸³ algorithms,¹⁸⁴ and data structures,¹⁸⁵ and the program structure or architecture.¹⁸⁶ The court retraces and maps each step the programmer took to

¹⁷⁷ *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.”).

¹⁷⁸ *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 112, 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).

¹⁷⁹ *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 Judge Learned Hand’s abstractions test, which was developed for plays and to isolate the “what the [work] is about” from the artistic elements).

¹⁸⁰ The main function of a program is the task that the computer performs and produces for the user. VERGARI & SHUE, *supra* note 11, at 2.

¹⁸¹ A source code is a program that is written in “a human-readable computer language.” VERGARI & SHUE, *supra* note 11, at 555.

¹⁸² 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).

¹⁸³ A program consists of modules, which perform a function. HUGHES & MICHOTM, *supra* note 41, at 22–23.

¹⁸⁴ 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).

¹⁸⁵ A data structure is “an organization of data” that holds data for which the algorithm can call on and has the computer execute those algorithms. Hamilton & Sabety, *supra* note 37, at 252.

¹⁸⁶ *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 various functions, which are performed by discrete modules, and how each of these modules interact with each other.”); *Comput. Assocs. Int’l v. Altai, Inc.*,

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.¹⁸⁷ The purpose of this step is to separate the ideas of the program from the expressions of the idea.¹⁸⁸

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,¹⁸⁹ or whether those protected aspects are "dictated by considerations of efficiency," the application of judicially-created defense to copyright infringement, the merger doctrine.¹⁹⁰ Once the court identifies each aspect as protectable or unprotectable, it disregards the unprotectable aspects.¹⁹¹ 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.¹⁹²

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.¹⁹³ After the court determines the

Inc., 982 F.2d 693, 707 (2d Cir. 1992) (providing a detailed "anatomical guide" to separating each part of the program).

¹⁸⁷ *Gates Rubber Co.*, 9 F.3d at 835; *Comput. Assocs. Int'l*, 982 F.2d at 707.

¹⁸⁸ *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 a plagiarist and a playwright's play, there comes a point where general statements from a play are no longer protected because a playwright could prevent the expression of general ideas).

¹⁸⁹ 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.").

¹⁹⁰ *Comput. Assocs. Int'l*, 982 F.2d at 707, 710-11. The merger doctrine is "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"; Faust, *supra* note 112, at 142.

¹⁹¹ *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); *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1167 (9th Cir. 1977).

¹⁹² *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.

¹⁹³ *Comput. Assocs. Int'l*, 982 F.2d at 710; Matthew P. Larvick, *Questioning the Necessity of Copyright Protection for Software Interfaces*, 1994 U. ILL. L. REV. 187, 199 (1994) (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).

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.¹⁹⁴

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.¹⁹⁵ The Copyright Act defines registration as “a registration of a claim in the original or the renewed and extended term of copyright.”¹⁹⁶ 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.”¹⁹⁷ 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.¹⁹⁸ 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.¹⁹⁹ Registration is

¹⁹⁴ 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 20, at 120.

¹⁹⁵ 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).

¹⁹⁶ 17 U.S.C. § 101 (2010).

¹⁹⁷ 17 U.S.C. § 410(d) (2010); *see* 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, who wants to protect his or her computer program, to deposit the source code, which is a human-readable language to the Copyright Office and, for specific types of programs, the Copyright Office has additional requirements).

¹⁹⁸ 17 U.S.C. § 410(d) (2010); 17 U.S.C. § 411(a) (2010).

¹⁹⁹ *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

only a prerequisite for a lawsuit, and courts decide the validity of the copyright.²⁰⁰

Pursuant to the holding of *Reed Elsevier, Inc. v. Muchnick*, registration is not a prerequisite for subject-matter jurisdiction.²⁰¹ However, courts can still dismiss a claim for copyright infringement on other grounds.²⁰² As a result, there is still a split in the circuits²⁰³ 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).²⁰⁴

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).

²⁰⁰ *Ward v. Nat'l Geographic Soc.*, F. Supp. 2d 429, 444-45 (S.D.N.Y. 2002); *Conley & Bryan*, *supra* note 95, at 578; *see also* *Bracey*, *supra* note 20, 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* Rita Marie Cain, *Timing Is Everything: Copyright Registration and Preregistration*, 88 J. PAT. & TRADEMARK OFF. SOC'Y 381, 384 (2006) (explaining how the Copyright Office could intervene in the infringement suit and argue the plaintiff's work does not have copyrightability).

²⁰¹ *Reed Elsevier, Inc. v. Muchnick*, 559 U.S. 154, 168 (2010); *see also* *Positive Black 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).

²⁰² *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).

²⁰³ *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, eighth, and District of Columbia, 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); *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).

²⁰⁴ *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

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.²⁰⁵ 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.²⁰⁶

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.²⁰⁷ In *La Resolana Architects v. Clay Realtors Angel Fire*, the plaintiff argued the defendant had used its drawings for a townhouse project and filed suit on November 20, 2003.²⁰⁸ 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.²⁰⁹

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.²¹⁰ Both the applicant and the Copyright Office have to take action to meet the registration

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).

²⁰⁵ *Kernel Records*, 694 F.3d at 1302; *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); Cain, *supra* note 200, 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).

²⁰⁶ 2 NIMMER & NIMMER, *supra* note 35, at § 7.16[B][3][b][i].

²⁰⁷ *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).

²⁰⁸ *La Resolana Architects*, 416 F.3d at 1197.

²⁰⁹ *Id.* at 1197–98.

²¹⁰ *Id.* at 1200.

requirements.²¹¹ The language of the statute does not suggest registration is met when the plaintiff receives confirmation from the Copyright Act of the application.²¹² Instead, the Copyright Office has to ascertain the validity of the application.²¹³ 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.²¹⁴

Then, the Tenth Circuit looked at the interpretation of section 410(a).²¹⁵ This section also requires the Copyright Office to take affirmative steps, such as examining, registering, and then issuing the certificate of registration.²¹⁶ However, section 410(a) does not suggest only filing an application is sufficient to register a work.²¹⁷

Once the Copyright Office approved the plaintiff's application, the issue was whether the plaintiff was required to have a certificate of registration.²¹⁸ 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.²¹⁹ The court's subject-matter jurisdiction depends on whether registration occurred and not on the issuance of the certificate.²²⁰

Finally, the Tenth Circuit provided the statute requires registration before filing a suit because Congress provided incentives and remedies to plaintiffs who registered.²²¹ 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.²²²

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.²²³ The plaintiff filed its application on May 5, 1986 and its suit on

²¹¹ *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).

²¹² *La Resolana Architects*, 416 F.3d at 1200.

²¹³ *Id.*

²¹⁴ *Id.* at 1201.

²¹⁵ *Id.*

²¹⁶ *Id.*

²¹⁷ 17 U.S.C. § 410(b) (2010); 17 U.S.C. § 408 (2010); *La Resolana Architects*, 416 F.3d at 1201.

²¹⁸ *La Resolana Architects*, 416 F.3d at 1202.

²¹⁹ *Id.*

²²⁰ *Id.*

²²¹ *Id.* at 1204; *see* 17 U.S.C. §§ 504, 505 (2010) (providing copyright statutory damages under section 504 and attorney fees under section 505).

²²² *La Resolana Architects*, 416 F.3d at 1207.

²²³ *M.G.B. Homes, Inc. v. Ameron Homes, Inc.*, 903 F.2d 1486, 1487 (11th Cir. 1990),

July 3, 1986.²²⁴ The district court stated the plaintiff failed to register its work for initiating a suit; therefore, the court lacked subject-matter jurisdiction.²²⁵ 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.²²⁶ 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.²²⁷ The Eleventh Circuit stated, once the plaintiff received the certificate of registration, it could have filed a new lawsuit.²²⁸

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.²²⁹ However, it did not rule on the issue of when registration has occurred under section 411(a) of the Copyright Act.²³⁰ As a result, circuits that choose the registration approach may still dismiss the suit on other grounds.²³¹

In *Fourth Estate Public Benefit Corporation v. Wall-Street.com, L.L.C.*, 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.²³² The Eleventh Circuit stated the registration application is voluntary, but Congress created incentives to register.²³³ Furthermore, the statute requires the Copyright Office to take steps when evaluating the application.²³⁴ The plaintiff has to take action, such as making

overruled by *Reed Elsevier, Inc. v. Muchnick*, 559 U.S. 154 (2010).

²²⁴ *Id.*

²²⁵ *Id.*

²²⁶ *Id.* at 1489.

²²⁷ *Id.*

²²⁸ *Id.*

²²⁹ *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).

²³⁰ *Reed Elsevier, Inc.*, 559 U.S. at 171; Kennedy, *supra* note 20, at 341.

²³¹ *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 non-compliance 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 the lower court's grant of summary judgment due to failure to apply for registration prior to initiating a copyright lawsuit).

²³² *Fourth Estate Pub. Benefit Corp. v. Wall-Street.com, L.L.C.*, 856 F.3d 1338, 1339, *petition for cert. filed*, 86 U.S.L.W. 3330 (U.S. Oct. 13, 2017) (No.17-571).

²³³ *Fourth Estate Pub. Benefit Corp.*, 856 F.3d at 1339.

²³⁴ *Id.* at 1340.

copies of his or her work, making a deposit with the Copyright Office, and filing the application.²³⁵ 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.²³⁶ 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.²³⁷

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.²³⁸

Under *Apple Barrel Products, Inc. v. Beard*, the plaintiff filed suit over the infringement of a county music program.²³⁹ The plaintiff filed the application but failed to register it.²⁴⁰ The Fifth Circuit reasoned the plaintiff did not need to possess a certificate of registration.²⁴¹ 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.²⁴² 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.²⁴³

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.²⁴⁴ The plaintiff asserted it met the registration requirement by receiving confirmation of the application from the Copyright Office before filing a suit.²⁴⁵ The defendant asserted the Copyright Office had to issue a certificate of registration.²⁴⁶

The Ninth Circuit looked at the plain language of section 411(a) of the

²³⁵ *Id.* at 1341; see 17 U.S.C. § 408(a), 410(a), (b) (2010).

²³⁶ *Fourth Estate*, 856 F.3d at 1341; see 17 U.S.C. § 410(a) (2010).

²³⁷ *Fourth Estate*, 856 F.3d at 1342.

²³⁸ *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).

²³⁹ *Apple Barrel Prods.*, 730 F.2d at 386.

²⁴⁰ *Id.* at 386–87.

²⁴¹ *Id.* at 386.

²⁴² *Id.* at 387.

²⁴³ *Capitol Audio Access, Inc. v. Umemoto*, 980 F. Supp. 2d 1154, 1157 (E.D. Cal. 2013).

²⁴⁴ *Cosmetic Ideas, Inc. v. IAC/InteractiveCorp*, 606 F.3d 612, 614 (9th Cir. 2010).

²⁴⁵ *Id.* at 614.

²⁴⁶ *Id.*

Copyright Act, but the section provided no guidance for interpreting registration.²⁴⁷ Then, the Ninth Circuit looked at the whole statute, sections 408 to 412, which discuss copyright registration, and determined the whole statute was ambiguous.²⁴⁸ The Ninth Circuit noted sections 410(a) and 411(a) require the Copyright Office to take some steps in registration.²⁴⁹ 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.²⁵⁰ Based on section 411(a), the Court said Congress intended not only the delivery of the application but also registration process included refusal or acceptance of the application.²⁵¹ 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.²⁵² Nevertheless, the plain language of the statute did not persuade the court.²⁵³ The Ninth Circuit examined the amendments to the Copyright Act including the provision stating registration is optional.²⁵⁴ 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.²⁵⁵

The Ninth Circuit reasoned the application view was the better approach based on policy considerations.²⁵⁶ 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.”²⁵⁷ 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.²⁵⁸

Pursuant to section 411(a), the Ninth Circuit held the plaintiff does not have to wait for the Copyright Office to approve or reject the application.²⁵⁹ In addition, the Court stated once the plaintiff receives the certificate, the registration refers to the data of the application.²⁶⁰

²⁴⁷ *Id.* at 616; *see* 17 U.S.C. § 411(a) (2010).

²⁴⁸ *Cosmetic Ideas*, 606 F.3d at 617–18; *see* 17 U.S.C. §§ 408–412 (2010).

²⁴⁹ *Cosmetic Ideas*, 606 F.3d at 617; *see* 17 U.S.C. § 410(a), 411(a); 17 U.S.C. § 411(a) (2010).

²⁵⁰ *Cosmetic Ideas*, 606 F.3d at 617.

²⁵¹ *Id.*; *see* 17 U.S.C. § 411(a) (2010).

²⁵² *Cosmetic Ideas*, 606 F.3d at 617; *see* 17 U.S.C. § 408(a), 411(a) (2010).

²⁵³ *Cosmetic Ideas*, 606 F.3d at 617.

²⁵⁴ *Id.*

²⁵⁵ *Id.* at 619.

²⁵⁶ *Id.* at 619–20.

²⁵⁷ *Id.* at 619.

²⁵⁸ *Id.* 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.

²⁵⁹ *Cosmetic Ideas*, 606 F.3d at 621.

²⁶⁰ *Id.*

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. 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. 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.

Some dissenters argue providing more protection to programs would hinder innovation and create monopolies.²⁶¹ Granting protection to merged elements of a program would “impede progress in the arts” and would be “contrary to the goals of copyright.”²⁶² 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.²⁶³ The purpose of copyright law is to promote and reward creative work pursuant to Article I of the Intellectual Property Clause, providing the public to benefit from various creations.²⁶⁴ 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.²⁶⁵

Opponents against more copyright protection for programs fail to understand

²⁶¹ *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 44 (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).

²⁶² *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).

²⁶³ Samuelson, *supra* note 26, at 741; Ambrose, *supra* note 108; see, e.g., *Apple Comput., Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1023, 1033, 1047 (N.D. Cal 1992) (reasoning that the plaintiff's work purely served a “functional purpose” and, based on the copyright law principles, the work received no protection).

²⁶⁴ U.S. CONST., art. 1, § 8, cl. 8; *Comput. Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 696 (2d Cir. 1992); Samuelson, *supra* note 261, at 1272; 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).

²⁶⁵ Samuelson, *supra* note 158, 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 155, 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.”).

the nature and design of the program.²⁶⁶ 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.²⁶⁷

The intent of the copyright law is to balance innovation with public welfare without burdensome requirements.²⁶⁸ Promoting innovation and maintaining public welfare can occur only if Congress understands the process of creating a program's main function as a whole.²⁶⁹ 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.²⁷⁰ The merger doctrine is an outdated, judicially-created concept that fails to consider the programmer's perspective and disregards the intent of the copyright law. 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.

Some courts assert the registration approach is in alignment with the text and the history of the Copyright Act.²⁷¹ 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.²⁷² Instead, the office has to determine the validity of the copyright protection.²⁷³ The language of the Copyright Act

²⁶⁶ Miller, *supra* note 10, at 1047; Mislow, *supra* note 57, at 778, 803; *see also* VERGARI & SHUE, *supra* note 11, at 510–11 (stating how the ever-changing role of technology has outpaced the legal realm).

²⁶⁷ Shaeffer, *supra* note 11, at 341, 345, 348, 368; Dunn, *supra* note 41, at 533 (treating the computer instructions and screen displays together provides the programmer with protection and certainty).

²⁶⁸ 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); *see also* Astle, *supra* note 18, 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).

²⁶⁹ Martyniuk, *supra* note 10, at 1369; Clapes et al., *supra* note 50, at 1545.

²⁷⁰ McGahn, *supra* note 176, at 132; Astle, *supra* note 18, 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 47, at 531.

²⁷¹ Fourth Estate Pub. Benefit Corp. v. Wall-Street.com, L.L.C., 856 F.3d 1338, 1340–42 (11th Cir. 2017); La Resolana Architects v. Clay Realtors Angle Fire, 416 F.3d 1195, 1200 (10th Cir. 2005), *overruled by* Reed Elsevier, Inc. v. Muchnick, 559 U.S. 154 (2010).

²⁷² Kernel Records Oy v. Mosley, 694 F.3d 1294, 1301–02 (11th Cir. 2012); *see* 17 U.S.C. § 410(a) (2010).

²⁷³ Gaiman v. McFarlane, 360 F.3d 644, 654–55 (7th Cir. 2004).

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.²⁷⁴

Proponents of the registration approach fail to understand the programmer's situation. 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.²⁷⁵ 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.²⁷⁶ When Congress required registration for lawsuits, the intent was to make it optional.²⁷⁷ Furthermore, courts do not dismiss the requirement of registration for filing a lawsuit; instead, they suggest registration requires fewer formalities.²⁷⁸

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.²⁷⁹ 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.²⁸⁰ The application view protects the programmer from the infringer having any unfair advantage and advances "the interests of justice" and "judicial economy."²⁸¹

The solution for providing programmers sufficient copyright and litigation protection is two-fold. 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. At the end of the section 102(b) of the Copyright Act,

²⁷⁴ *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).

²⁷⁵ Hogan, *supra* note 19, at 849-50; see 17 U.S.C. § 412 (2010) (explaining how a copyright infringement suit must be brought within three years).

²⁷⁶ *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 20, at 141.

²⁷⁷ Hogan, *supra* note 19, at 846.

²⁷⁸ Bracey, *supra* note 204, at 139 (requiring both the application and the deposit to be sent under both approaches).

²⁷⁹ *Cosmetic Ideas*, 606 F.3d at 619; Kennedy, *supra* note 20, at 344.

²⁸⁰ Cain, *supra* note 200, at 385.

²⁸¹ *Int'l Kitchen Exhaust Cleaning Assocs. v. Power Washers of N. Am.*, 81 F. Supp. 2d 70, 72 (D.D.C. 2000); *see also* Bracey, *supra* note 20, 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); Hogan, *supra* note 19, at 866 (adopting the registration approach would lead to an imbalance between "creative ownership and cultural process."); 2 NIMMER & NIMMER, *supra* note 35, at § 7.16[B][3][b][iii] (stating the issue with backdating the certificate when the statute of limitations has run out).

Congress should add a line stating, “For purposes of this subsection, computer programs whose ideas and expressions merge are considered protected.” 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. 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 step, the merger doctrine would not apply, which would have removed the protected aspects of the program. 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.²⁸²

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. 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.

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. Congress will make it easier not only to register a work without having to wait, but also place others on notice. 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.”²⁸³ 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 public will have access to a national registry.

²⁸² 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).

²⁸³ Astle, *supra* note 18.

