SSL Ethics Essay

Copyright is a subject about which many programmers lack the proper knowledge, and so did I prior to researching for this paper. When it comes to programming copyrights, there is much that isn't understood by general circles, including clear definition of what copyright really means, delineation of license agreements and its types, and situations that can come as a result of a poor understanding of copyright practices. All this will be discussed within the lens of ethics: the ACM Code of Ethics and the IEEE Code of Ethics, as well as my own personal ethics as a Christian.

Copyright, as it specifically applies to programming and program creation, is defined by the US legal system. It states that copyright applies to the "copyrightable expression embodied in the program," not the "functional aspects of a computer program, such as the program's algorithms, formatting, functions, logic, or system design" (*Circular 61: Copyright Registration of Computer Programs*). Copyright protection can come under a number of different defenses in the form of license agreements that set the limits of accessibility. In other words, as was discussed in class, as soon as you write and save some code, it is yours and exclusively copyrighted (unofficially) ("No License"). To get it "officially" copyrighted, you would have to apply for that in a registration process. However, there are parts of a program that aren't copyrightable, like what I personally call "generic code" – sections of code that are essential foundations to larger programs. One example of this in C++ would be the basic format of a for loop. It's a basic format used repeatedly, oftentimes line for line when used to check elements of an array. It is specifically said that this sort of thing cannot be copyrighted. There is a certain type of distribution that a large number of programmers use called open source. Open source is a

form of publishing programming to where it is meant to be received by others in some capacity, depending on the license agreement chosen.

License agreements are a crucial part of creating open source programs. Without the license agreement, you are, by default, preventing others from actually being able to interact with your code – the entire purpose of open source programming ("No License"). The license allows others to interact with your code in different ways depending on what license agreement is made ("No License"). When it comes to choosing a fair license agreement for my code, I would have to consider which to choose based on the circumstances. There isn't any one license agreement that I believe I should go by all the time. If I want to keep it as my own for a time, I would hold off on choosing a specific license until I would like to release it (like if I were a teacher for example). If I want to just make it relatively open, but also keep it to where the project has a clear line of progression and shows the original source code, then I would want to use the GNU GPLv3 license ("Choose an Open Source License"). If I just wanted to post something for essentially anyone to use as they wish, or if I just wanted to quickly choose a license for a project, the MIT license is the one to choose ("Choose an Open Source License"). There are many other license agreements with variations in availabilities that are far too numerous to go through one-by-one. Personally, I like the idea of being able to see the original source code alongside the modified version. It makes for both efficient sourcing, proper crediting, and it allows for sharing all the same as a more public license like the MIT. For the same reasons, I also believe it to be the fairest because it isn't too restrictive while also giving credit where it is due. However, the short and simple answer to what license I would use, in the end, is "it depends."

When it comes to reusing code from the internet, things can get complicated ethically for programmer. With open source projects like on Github where you can get a license, the terms set in place are relatively clear. However, when something is just posted on the Internet on a website like stackoverflow.com, the ethics are more in question. As for my personal experience with this, my numerous programming classes have told me to just put my source url in a comment next to where I used it. Of course, we are going to need to take a closer look at this technical issue with a more professional lens as well as my own personal ethical opinion on the issue. Two of the major professional codes of ethics are the ACM Code of Ethics and Professional Conduct and the IEEE Code of Ethics. These two, while similar in some ways, are significantly different, which influences my choice in the one I will choose to judge actions and behavior as a programmer ethically. Looking at the ACM and IEEE codes, it is clear that the ACM principles put a greater focus on aspects of computer science, both specifically and generally ("ACM Code of Ethics and Professional Conduct"). The IEEE principles are not for programmers specifically, and instead are made to be generally relevant to a wide range of technological fields ("IEEE Code of Ethics"). In regards to copyright issues and the ethics relating to it, the ACM is very specific about "respect[ing] the work required to produce new ideas, inventions, creative works, and computing artifacts": recognizing that both legally and ethically, the use of someone else's code should, depending on the copyright permissions and such, give proper credits to the original creators or not copy it at all ("ACM Code of Ethics and Professional Conduct"). At the same time, it notes that creators shouldn't just oppose all requests to use their code since it can be used for the good of society ("ACM Code of Ethics and Professional Conduct"). Also, one shouldn't falsely claim public code as their own copyrighted code either ("ACM Code of Ethics and Professional Conduct"). With the IEEE, the copyright issues and ethics aren't really referenced

in specifics at all (“IEEE Code of Ethics”). It only has general guidelines of respecting others and to remain within the law in your professional life (“IEEE Code of Ethics”). Clearly, the ACM would be the way for me to judge ethical situations as a programmer. The IEEE is far too vague.

Now we return to the technical ethical issue concerning code from the Internet. When it comes to using sites like stackoverflow.com, the website and others like it have a Creative Commons license suited for what level of distribution and type of crediting they want (“The Legal Side of Open Source”). Stackoverflow.com specifically has a CC BY-SA 4.0 license that “allows reusers to distribute, remix, adapt, and build upon the material” if the creator is given proper credit (“Public Network Terms of Service”; “About CC Licenses”). Under the ACM code discussed previously, we would follow this policy and give credit to whatever we sourced our code from on stackoverflow.com. Before I learned about crediting answers from websites like stackoverflow.com, I had my own personal ethic as a Christian. Proverbs 12:24 states that “Diligent hands will rule, but laziness ends in forced labor” (Proverbs 12:24, NIV, 2011). What I would do is try my best to solve an issue that I’m having trouble with, exhausting all other resources available like textbooks, notes, and online lectures. If I couldn’t find anything in there, then I would go to the Internet for help. If I didn’t understand the answer from the Internet but used it anyways because it works, I would do further research to figure out what it meant and/or ask my professor later. (I still do a similar method to this today, but I just credit the original coder as well.) Of course, that didn’t align with the ACM since I didn’t give proper credit, and it is unacceptable legally speaking, but I have updated my personal ethic since then. Listing the source of what inspired my code does follow with my ethics because Leviticus 19:11 states “Do not steal. Do not lie. Do not deceive one another,” which resonates strongly with the idea of not

passing someone's work as your own, even if unintentionally done (Leviticus 19:11, NIV, 2011). However, while this may work for a classroom situation, this technical choice does not have the same impact for an actual job. In fact, this issue is something that neither the ACM nor my personal ethic covers the bases for. While you may legally be able to copy the code and give credit, what you can't trust is if what the person posted is actually their code ("Do I Have to Worry about Copyright Issues for Code Posted on Stack Overflow?"). With a company program or a program for a product, you would have to prove that you got the code from stackoverflow.com and not somewhere else ("Do I Have to Worry about Copyright Issues for Code Posted on Stack Overflow?"). The person you copied could have potentially gotten that from anywhere else, and that would lead to disarray in the project ("Do I Have to Worry about Copyright Issues for Code Posted on Stack Overflow?"). What to take from the situation is never to directly copy code from stackoverflow.com or any website for that matter in this sort of situation; instead, figure out how to make your own version that works for what you need it to ("Do I Have to Worry about Copyright Issues for Code Posted on Stack Overflow?"). While the ACM is a helpful set of guidelines for programmers to follow ethically, it doesn't necessarily cover all circumstances and situations.

A wide variety of subjects related to copyright continue to perplex programmers. It is now known what the US legal definition of copyright is as well as what a license agreement is. We looked at a couple of the many license agreements available, seeing that each was useful depending on what is desired. The ACM and IEEE ethics codes were compared, as was my personal Christian ethics. Lastly, I used the ACM and my own ethics to determine what should be done with code found on the Internet. Overall, when it comes to copyright, programmers have to be far more cautious, and they should be sure that check the copyright policies of whatever

websites or programs they interact with. Knowing all of these definitions and legalities are ultimately important for their jobs, and becoming familiar with them will make programmers' lives easier in the future.

Works Cited

"About CC Licenses." *Creative Commons*, 22 May 2020, https://creativecommons.org/about/cclicenses/. Accessed 3 October 2021.

"ACM Code of Ethics and Professional Conduct." *Acm.org*, 2018, https://www.acm.org/code-of-ethics. Accessed 2 October 2021.

"Choose an Open Source License." *Choose a License*, https://choosealicense.com/. Accessed 3 October 2021.

Circular 61: Copyright Registration of Computer Programs. https://www.copyright.gov/circs/circ61.pdf. Accessed 3 October 2021.

"Do I Have to Worry about Copyright Issues for Code Posted on Stack Overflow?" *Meta Stack Exchange*, https://meta.stackexchange.com/questions/12527/do-i-have-to-worry-about-copyright-issues-for-code-posted-on-stack-overflow. Accessed 3 October 2021.

"IEEE Code of Ethics." *IEEE.org*, https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 2 October 2021.

New International Version Bible (2011). Zondervan.

"No License." *Choose a License*, https://choosealicense.com/no-permission/. Accessed 3 October 2021.

"Public Network Terms of Service." *Stack Overflow*, https://stackoverflow.com/legal/terms-of-service/public#access. Accessed 3 October 2021.

"The Legal Side of Open Source." *Open Source Guides*, https://opensource.guide/legal/ .

Accessed 2 October 2021.