Education Libraries, 44:1 (2021)

Research Article

Coding in School Libraries: Considering an Ethical Approach

JENNA KAMMER[1]
Assistant Professor, University of Central Missouri, Warrensburg, MO, USA

LAUREN HAYS[2]
Assistant Professor, University of Central Missouri, Warrensburg, MO, USA

Abstract

This overview demonstrates the need for school librarians to consider supplementing coding instruction with ethical discussions. School librarians are increasingly incorporating coding into library instruction through play, tutorials, collaboration in content areas, and design thinking projects. To enhance students' ethical decision making and empathy for others, school librarians can also incorporate ethical decision-making into coding activities. In this article, the authors explain why this is important, and present an ethical discussion model. These strategies may help school librarians work towards teaching social justice issues as part of coding programming.

Keywords: coding, school libraries, ethics

[1] **Jenna Kammer**, Ph.D., M.L.S., M.A. Ed., is an assistant professor at the University of Central Missouri. She teaches library science. Prior to working as a professor, Dr. Kammer was an instructional designer and reference librarian.

[2] **Lauren Hays**, Ph.D., M.L.S., is an assistant professor at the University of Central Missouri. She teaches educational technology. Prior to working as a professor, Dr. Hays was an academic librarian.

Introduction

In the book, *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*, Eubanks (2018) explains how automated systems have worsened inequalities for the poor through profiling, surveillance, and punishment. One of the reasons for this is that engineers, programmers, and data scientists themselves have pre-existing bias about poverty (Eubanks, 2018). Bias in code occurs when programmers make choices, choose data, and select metrics without considering the ethical implications. The choices, data, and metrics are not neutral and need to be fully understood to lessen the harm that can be done. One way to reduce programmer bias is for computing professionals to consider the social and economic implications of their designs and approach their work with a "philosophy of non-harm" (Eubanks, 2018, p. 212), something similar to a Hippocratic Oath for those working with computer systems. Eubanks' call to action is an opportunity for school librarians to consider how to teach coding with this same philosophy in mind.

As educators, we were particularly inspired by Eubank's book and its message. We felt it addressed critical concerns for preparing future educators that are often overlooked in technology education. We sought to understand more about the problem of bias in technology, as well as identify how these concerns are addressed in practice. In this paper, an overview of coding in school libraries is provided, with an emphasis on identifying opportunities for school librarians to include ethical practice when working with young programmers. For school librarians in particular, the intersection of coding and information ethics is highly relevant, and the American Association of School Librarians (2018) have established goals related to the ethical use of information.

School Libraries and Coding

Libraries that serve children increasingly offer opportunities for students to experience coding, with coding and maker activities serving as the top two tech-related activities in the library (School Library Journal, 2019; Subramanium et al., 2019). Harrop (2018) describes how libraries of all types offer coding activities for youth as part of makerspaces, coding classes, coding clubs, or code-a-thon events. Libraries may offer coding to fill a gap in computer science programming in their community or school, or include it because it is an interactive, enjoyable learning experience for students (Martin, 2017). In some schools, the library may also be the only location where students can access coding activities. Many school libraries include coding as part of their STEM programming, particularly at the elementary level where coding supports digital literacy and provides equitable opportunities for students to learn a technical skill useful in the 21st century workforce (Dadlani & Todd, 2015).

School librarians increasingly integrate technology into learning, with many serving as technology leaders in their schools. Wine (2016) explains that school librarians work in collaboration with technology specialists and may even serve as the technology specialist if there is not another professional in that role. The School Library Journal's (2019) *Technology Report* also shared data that demonstrated that school librarians identify as technology experts, spearheading coding initiatives and often being responsible for technology usage or instruction within the school. In addition, the

American Association of School Librarians (AASL) (2019) has responded to the influx of school librarian-led coding initiatives by creating a crosswalk that aligns the AASL standards with Code with Google's CS First curriculum. The crosswalk explains that within each of the AASL Shared Foundations, there are one-hour and multi-day activities available in the CS First curriculum that school librarians can employ to meet learning goals (AASL, 2019).

Coding in school libraries is a fun and interactive experience for students where they can practice in a low-stakes environment, often free from assessment. However, school librarians (and many other teachers of computer science in public schools) are not computer scientists and have often received no specific training or education in the area (Webb et al., 2017). Whereas professional computer programmers follow a code of ethics that describes how they should act responsibly and adhere to certain practices, behaviors, and attitudes (Association for Computing Machinery [ACM], 2018), school librarians may be unfamiliar with these principles or how to integrate them into coding practice. In general, the ACM code of ethics used by computer professionals explains that they should avoid harm, be honest and fair, avoid discrimination, and respect privacy and confidentiality (Association for Computing Machinery, 2018). These principles are very similar to concepts from information ethics, a branch of ethics that is tightly woven into school library standards, and provides a framework for considering privacy, confidentiality, censorship, moral agency, and issues related to democracy and the public good (Oltmann, 2018).

To understand more about the integration of computer ethics in coding pedagogy for school librarians, literature that discussed the practice of teaching coding in schools was examined for the purposes of this paper. Practical examples of coding with students from teachers and librarians whose primary role was *not* to teach computer science were sought. Education and library databases were searched with keywords such as "school librarians and coding" or "coding and schools." Results were delimited to exclude literature unrelated to K-12 schools, and also to exclude literature related to teaching computer science courses. There was a wealth of existing information that explains the issue of technology bias, and a wealth of information on computer science curriculum in K-12 classrooms and libraries, but with little overlap between the two. Therefore, both the problem of technology bias in coding, as well as the practice of teaching coding in schools was examined. Finally, we make recommendations for integrating existing models of ethical decision-making into conversations with students around their coding projects as a strategy for teaching K-12 students about the impacts of computing.

The Expanding Coding Curriculum

Karsenti (2019) explains that *coding* is the word used to describe the process of creating code without formal computer science training, and a *programmer* is the job title used to describe a position that employs professionals with training in specific coding languages. Both terms will be used respectively: coding, as an informal activity, and programming, as the more formal and professional version that happens within computer science curriculum. While students can code without the intention of becoming a programmer, coding programs are often implemented with the hopes that some young students from diverse backgrounds will indeed become programmers, enter the computer science profession, or use programming skills in their places of employment (Martin, 2017).

Computer science curriculum is expanding in K-12 schools as new computer standards are developed within each state (Google Inc & Gallup Inc, 2016). Coding is one component of a computer science curriculum. Harrop (2018) explains that coding involves writing a sequence of commands in a specific programming language. For librarians, the strongest incentive to offer coding opportunities to children is to support lifelong learning so that they may enter careers related to computer science if they choose or use coding skills in other fields. While Harrop (2018) suggests that librarians offer coding opportunities in the library to prepare students for careers that require a coding background, such as website development, engineering, or software design, others say that coding skills are useful in any career because they help to develop computational thinking as well as support digital literacy (Hutchison et al., 2016). For example, Hutchison et al. (2016) describes how coding can be a way of self-expression for students, and that a student fluent in a coding language can tell stories, be creative, or collaborate in a variety of ways.

In countries such as Finland, England, and South Korea, computer science is a compulsory subject (Duncan & Bell, 2015; Uzunboylu et al., 2017). In the United States, 39 states have, or are adopting, computer science curriculum for public schools (2019 State of Computer Science Education, 2019). Furthermore, there are computer science initiatives already in place in many countries, such as Australia and New Zealand. These initiatives seek to integrate computer science curriculum into existing curriculum initiatives (Webb et al., 2017). The K-12 Computer Science Framework (n.d.) was developed through a coalition of national organizations to support the development of computer science standards at the state and district levels. In the United States, computer science curricula are developing at the state level through the efforts of organizations like Project Lead the Way, Code.org, the Computer Science Teachers Organization, Google, and the National Math + Science Initiative (K-12 Computer Science Framework, n.d.).

Teaching coding is a term that is synonymous with programming education, although the practice of *teaching coding* may have different meaning to different teachers who have varying levels of experience or are using different standards (Duncan et al., 2014). For example, coding may be integrated into content or specialization areas, often by educators whose computing knowledge is self-taught or learned in professional development for educators. In elementary schools, teaching coding may refer to activities where children can integrate play while learning coding independently or use self-paced coding tutorials (Kalogiannaikis & Papadakis, 2017). At the secondary level, teaching coding refers more to the teaching of programming languages and is often taught within full computer science courses by teachers with more formal training (Duncan et al., 2014).

The Problem: Technology Bias within the Coding Industry

An increasing body of research suggests that technology bias has significant social ramifications for those who are already socially disadvantaged (see Dadlani & Todd, 2015; Howard & Borenstien; Johnson, 2020; Naidoo & Sweeney, 2015; Noble, 2013). Digital media scholars, like Noble (2013), describe how technology companies prioritize commercial interests over fair representation, thus creating technological blind spots that fail to represent specific groups of people by race, gender, or religion. However, educators can prepare students to think critically about existing inequalities in technology systems by integrating critical technology perspectives. Naidoo and Sweeney (2015) suggest that intentional integration of critical theories in instruction can help students identify social inequities and begin to develop solutions for them.

Fiesler et al. (2020) calls for an increase in computer ethics pedagogy as more teachers teach computer science principles in their K-12 classrooms. They explain that learning to use technology should involve learning how to avoid biased or harmful practices. One reason for this is that in-group bias is a general problem in computer programming, a field dominated by white men (Levin, 2019). Programmers may unintentionally embed their own biases in code they write. In other words, someone may be more likely to represent the social group that they are part of when making decisions (Bruneau et al., 2017). In-group bias has been studied extensively to understand how people behave in groups. One of the most famous examples of in-group bias is the Robbers Cave Experiment (Sherif et al., 1961). In this experiment, boys at a summer camp were placed in small groups and separated from each other. Their interactions with rival groups were studied by researchers. The researchers found that the boys would consistently show favoritism towards their own group (Sherif et al., 1961).

Programmers and designers of artificial intelligence (AI) within many tech corporations are working to address in-group bias because of a general lack of diversity across the field (Hern, 2016). Algorithms, followed by or created by AI, impact what news we see, what entertainment we partake in, who we spend time with, our access to credit, and our access to public services (Osoba & Welser, 2017). As machine learning continues to influence the algorithms that we interact with daily, those algorithms continue to learn human biases, because the data they use to learn (social media, etc.) is biased (Osoba & Welser, 2017). Articles in popular media, such as *The Guardian*, have brought attention to this and have recommended that artificial intelligence ethics boards attempt to address some of these issues and set best practices to be followed (see Hern, 2016; Levin, 2019).

Bias exists in code and algorithms in many different forms. Implicit bias happens when cognitive biases impact a person's beliefs or actions towards other people (Johnson, 2020). In computing, implicit bias can become algorithmic, or machine embedded, when it mimics human implicit bias (Johnson, 2020). Different populations experience the negative impact of machine embedded biases. Villasenor (2019) identified four key challenges with bias in artificial intelligence (AI): bias built into data, AI-induced bias, teaching AI human rules, and evaluating cases of AI suspected bias. One example of AI-induced bias, described by Villasenor (2019), relates to software making loan decisions. Individuals from middle-income neighborhoods are more likely to make higher incomes than those from lower-income neighborhoods and thus more frequently receive approval for loans. Over time, AI consuming this data could start giving individuals in middle-income neighborhoods loans because of their zip code even if they do not have as many assets as an individual from a lower-income neighborhood (Villasenor, 2019). Additionally, AI can have a hard time distinguishing when something is a culturally accepted practice such as tipping or when it is an unlawful practice such as bribing. Finally, things that can appear to be biased are not always that, but untangling AI's reasoning can prove challenging as the code develops on its own through machine learning (Villasenor, 2019).

New technologies also impact individuals in ways unforeseen by those who create tools and those who implement them. Eubanks (2018) explains how new technology is keeping some members of Los Angeles' homeless community on the street because they do not qualify for housing programs based on algorithmic data. Similarly, in Indiana, a state technology system denied coverage to families eligible for food stamps and healthcare for non-compliance without explaining how to get in compliance, thus forcing families to reapply for benefits (Eubanks, 2018). The purpose of these new tools was to streamline processes and save money. Instead, it created more disadvantages among vulnerable populations.

Google search results, based on AI algorithms, are another clear example of technological bias. After years of research, Noble (2018) found that search engines discriminate against women of color and instead elevate whiteness. Discrepancies in search results may not be immediately apparent, but the returned content varied greatly when terms such as *black girls* and *white girls* were searched (Noble, 2018). Noble (2018) found that when searching for *black girls* the top search results were pornography sites. In this way, algorithms can promote racism, and the results have direct implications on how search engine users perceive the world. Johnson (2020) also explained how searchers are more likely to search for *is my son gifted* than *is my daughter gifted*, inadvertently training AI to mimic implicit bias patterns.

Algorithms are embedded into our lives in ways that are almost impossible for us to extricate ourselves from. If we engage online in any way companies capture our data and then use that data to make decisions about us as consumers, investors, and employees. This data gathering and use, and the algorithms behind it, mean we can see higher prices for products based on our purchasing history, be rejected for a loan because we are in marriage counseling, and be or not be considered for a job based on the words we use in our resumes (Pasquale, 2016).

These biases that exist in code and algorithms may not be intentional; but they do highlight the implicit biases and blind spots of the individuals developing the software. Too often, the people who create software come from similar backgrounds and do not have a strong understanding of how their work will be implemented, or how it will impact the lives of users. Additionally, the way algorithms manifest themselves in people's lives can be unknown to their creators. Many algorithms are built to learn from the data they collect; therefore, they can learn to categorize people in ways the programmers did not foresee, or specifically intend.

Coding Practices in School Libraries

Coding practices within libraries are often described in library professional publications. Current school library trade journals were examined in order to understand how coding is taught in school libraries. Four strategies for teaching coding were identified from the literature: teaching coding through play, the drag and drop tutorial, coding across the curriculum, and coding as part of design thinking projects.

Teaching coding through play. There are many coding toys available for children to learn coding while playing. Many libraries and schools have makerspaces that include robotics, coding applications, or 3D modeling. These spaces are often for experimenting or exploring and allow students to learn coding in an unstructured, fun way. Moorefield-Lang (2015) describes makerspaces as sources of exploration, that include "making, hacking, inventing, crafting, or 3D printing" (p. 108). An underlying theme for makerspaces is learning, and the combination of a social space with innovative tools and resources is an ideal learning environment (Willett, 2016). Essentially, students use their own problem-solving strategies to play with the coding toys and learn through a relaxed and fun format. Some of the tools that are commonly found in a school library makerspace are robots and other coding tools, such as Squishy Circuits, Makey Makey, or Osmo, (Lamb, 2016). Kafai et al. (2014) recommends that activities using these tools be done in groups to encourage young coders to think about coding in teams and from a community perspective.

The drag and drop tutorial. Learning to code using a tutorial is a common approach for introducing computer science to youth (Frederick, 2015; Smith, 2018). There are a variety of free tutorials designed to teach students to code, like Google's CS First, Scratch Jr., Hour of Code, and Code.org (Smith, 2018). These programs provide coding blocks that students can easily manipulate. They are heavily scripted and include instructions, lesson plans, and assessments. Teachers can use the lessons as is or adapt them. A program like CS First also includes a learning management system so that teachers can manage their students' activity and completion. In addition, CS First offers reflective opportunities meant to increase students' problem-solving abilities (Google for Education, n.d.). The Hour of Code program can be introduced in school libraries, outside of the classroom and other existing curriculum (Frederick, 2015). These programs allow for librarians to provide coding opportunities to students without having to know how to code themselves (Frederick, 2015). Colby (2015) shared experiences after conducting an Hour of Code event at a high school. Students with varying levels of coding experience completed tutorials that introduced them to computer programming (Colby, 2015).

Coding Across the Curriculum (STEAM projects). School librarians are instructional partners who work with other educators to collaborate to meet curriculum and information literacy goals. Coding can be integrated into an existing project in a content area to create a more meaningful coding project. Moura (2018) refers to this as "Coding Across the Curriculum", similar in language to Falkner and Vivian's (2015) technical report, *Coding Across the Curriculum: a Resource Review.* Falkner and Vivian (2015) present several ways that coding can be integrated into other subject areas. For example, in a geography lesson, students can draw a map and code a robot to visit different places or write if-then statements for a storytelling assignment in an English class. Hutchison et al. (2016) also recommend that coding be taught in combination with other common core standards. They provide an example of how *My Robot Friend*, a coding app (which is no longer available) in which students use an algorithm created with symbols to code a robot to stack cups, could be aligned with English Language Arts standards to create a more meaningful activity (Hutchison et al., 2016).

Design Thinking Projects. One way to teach more robust computational concepts is to integrate design thinking. Design thinking includes five stages of problem solving: empathize, define, ideate, prototype, and test (Cross, 2011). Cross (2011) explains that the first stage of *empathize* encourages students to research the needs of the user. This step is important for promoting ethics and avoiding bias. Instead of designing a product the way the student wishes it to exist, the student must learn what needs the user possesses. This process builds empathy in the student (designer).

Design thinking can also be used to teach coding by having students think backwards about developing an app (Kiang, 2015). Instead of learning to code first, students are asked to start with a problem, then work through it systematically to create an app that can solve the problem (Kiang, 2015). Another initiative that encourages students to think about coding from a design standpoint, is the ALA Ready to Code project. The ALA Ready to Code project is a collaborative initiative between Google and the American Libraries Association to integrate coding and computational thinking activities into library programming for youth (Subramanium et al., 2019). Subramanium et al. (2019) explains that in the Ready to Code initiative, Google provided funding for several different libraries to integrate coding projects into the library. Some of these projects utilize design thinking to encourage students to think like engineers. One of the examples shared on the Ready to Code website, is the Feathered Friends project (Visser, 2018). In the Feathered Friends project, a school district used connected learning, design thinking, and computational thinking to enhance their coding initiative, and older and younger students worked together in a client-engineer relationship (Visser, 2018). The older students took the role of engineer and interviewed younger students about their "problem" before designing a solution for them. The interviews consisted of learning more about the younger students, including their needs and use of the project, as well as how it would affect them (Visser, 2018).

Recommendation: Computer Ethics Pedagogy

Each of the strategies described above could include opportunities for the school librarian to have productive conversations around students' ethical coding practices. School librarians must make an intentional effort to include elements that require students to think outside of their own experiences and to understand how others perceive or are affected by an application. It is important for school librarians to understand that many of the existing coding tutorials and activities are more technical in nature and need to be enhanced to encourage students to think about the impact of coding. Jones (2016) suggests that ethical considerations should be embedded into student coding projects from the start. For example, a CS First tutorial is very well-designed for children to learn to code and execute tasks. It even includes reflective opportunities which support the problem-solving process. But as a standalone tutorial, it will not ask students to consider the broader social context. The tutorial needs supplementation by the school librarian to include opportunities for students to think outside of themselves and what they already understand about the world. A few simple ways to enhance a coding tutorial might be to allow students to engage in group discussions prior to coding, work in teams of diverse groups to create or test the code, or seek feedback from others after completing the project.

Computer ethics assumes that there are societal implications that affect core human values (life, happiness, freedom, security, etc.) when using computers (Moor, 1999). Specifically, Wiener (1954) speculated that the rise in computing and automated systems would lead to social and ethical inequities. At the time, Wiener used the term *cybernetics* to describe how machines would make decisions for humans and theorized that this would have implications for modern society. They recommended that those working with computers should consider the effects of automation on the individuals who will use it. In the 1950s, humans were still able to beat a computer at playing chess, however Wiener predicted that computers would soon be able to learn and then beat humans, with many implications to follow. As predicted, this kind of computer-driven decision making has since been used for war games, business, and other artificial intelligence operations (Bynum, 2017). Computer ethics frameworks ask that these programs be developed with care and concern for others.

Walter Maner further developed the concept of information ethics to specifically include computers and coined the term *computer ethics* (Bynum, 2001). Maner (2002) noted that computers made ethical situations more complex. To find a practical way to apply ethics to computing, Maner explored the domain of 'procedural ethics'. They found several heuristic procedures that could be used to guide ethical decision-making. One of these procedures is called The Ethical Decision-Making Model. This model was developed specifically for teenagers and is a procedure that allows students to consider conflict resolution in several ways:

1. Identify stakeholders.
2. Identify values: ethical and nonethical.
3. Ethical values trump nonethical values.
4. If two ethical values conflict, the one that produces the greatest good for the greatest number wins. (Maner, 2002, p. 349)

Integrating an ethical decision-making model into a classroom or existing lesson is often referred to as embedded ethics, or micro-insertion (Davis, 2006). Teaching ethics through an embedded ethics approach can improve students' ethical judgement (Davis, 2006). Specifically, Davis suggested that ethics can be integrated into a curriculum without making substantial changes. One way to do this may be to include case studies or scenarios that students can discuss (Kert et al., 2012). Kert et al. (2012) describes several scenarios that can be used for computer ethics education, like ethical dilemmas. These scenarios allow students to engage in decision making processes related to real life situations that arise because of computers. In one case, Kert et al. (2012) used the example of a security threat, in which a student creates an algorithm that can detect security vulnerabilities in a computer system. However, a hacked company traces the attack back to the school where the student resides (Kert et al., 2012). In this ethical dilemma, computer science students would consider the perspectives of the hacked company, the student, the university, and the professor, to determine the ethical issues (Kert et al., 2012).

School librarians in K-12 schools can also incorporate ethical discussions into their coding lessons. Even if students work individually on coding tutorials, conversations before, during, or after the lesson can encourage students to think about their projects from an ethical standpoint using an ethical framework. These conversations can be led by guided questions that do not have right or wrong answers and can encourage students to think more deeply about the ethical implications of their projects. Table 1 and Table 2 were created to provide examples of questions that school librarians can use to guide discussions about ethics in coding. The examples include core human values and were chosen to show how school librarians can guide discussions at different grade levels. Scenarios were constructed that incorporated coding activities a school librarian might experience, and questions were developed to encourage critical, ethical thinking related to the activity and within Maner's (2002) framework. These examples were inspired by Moor's (1999) seminal article on computer ethics that lists core human values. In the examples, ethical and nonethical values are included. Ethical values are those values that are universally accepted (e.g. honesty, integrity) while nonethical values speak to what individuals prioritize (e.g. financial stability, efficiency) (Moor, 1999).

Table 1. Guiding questions for ethical discussions for middle school students aligned with Maner's (2002) ethical decision-making model scenario 1.

Scenario: A group of middle students are working on a coding project. Their task is to design a technology that will help others. The group decides to write a code that will trigger a camera to record video of children playing on the playground so they can identify when a child has been bullied. The teacher wants to encourage students to think more critically about this idea.

Ethical Procedure	Guiding Questions (example)
Identify stakeholders.	Who is involved? Who will be affected? Who will be helped? Who will be hurt?
Identify values: ethical and nonethical.	What is the value of this project? What are the reasons we should create this project? What are the reasons we should not create this project? What will happen if…(e.g. a student is embarrassed about what is seen on the video? The camera records things not related to bullying? Someone is accused incorrectly of being bullied? What if someone is bullied and the camera doesn't catch it?)
Ethical values trump nonethical values.	Do any of these concerns fall within one of the core human values? (e.g. life, health, happiness, security, resources, opportunities, knowledge, privacy)
If two ethical values conflict, the one that produces the greatest good for the greatest number wins	Is there a way to modify this system to produce more good for more people?

Table 2. Guiding questions for ethical discussions for elementary students /aligned with Maner's (2002) ethical decision-making model scenario 2.

Scenario: Elementary students create their own superhero using code in a block-based programming platform. The teacher wants to encourage the students to consider empathy as they design their superhero.

Ethical Procedure	Guiding Questions (example)
Identify stakeholders.	Who is your superhero? What powers does your superhero have? Who else is in your superhero story?
Identify values: ethical and nonethical.	What does your superhero do to help people? Who does your superhero help? What are the reasons that someone would need your superhero?
Ethical values trump nonethical values.	Which of the core human values does your superhero help with the most? (e.g. life, health, happiness, security, resources, opportunities, knowledge, privacy) What do your friends find valuable about your superhero?
If two ethical values conflict, the one that produces the greatest good for the greatest number wins	Who else could your superhero help that you did not think of before? Is there anything else your superhero can do with their powers to help more people?

Discussion

We began this paper with two goals. The first was to understand more about ethical concerns in programming, and their implications on society. The second was to understand more about how school librarians teach coding in order to identify ways to introduce ethical concepts related to technology applications. What we found was that there is very little research and even less practical application addressing computing ethics at the K-12 level. In addition, little research on the coding practices of school librarians has been published. There are, however, many current papers related to coding in school libraries, that describe engaging activities and have suggestions for school librarians who wish to build a coding program in their school. Of all the examples found during this review, only the Feathered Friends project (Visser, 2018), included discussions with students related to ethical decision making.

In K-12 education, coding and computer science is a growing content area with new standards and guidelines emerging around the world. If children learn to code with others in mind at a young age, they will be well-prepared to work as programmers or in other jobs that require them to consider the impacts of their work on those affected. There are several common ways to teach code, including through games, block-based programming, and through content-based projects. As such, an ethical decision-making process, as recommended within the field of computer ethics, may allow librarians to supplement basic coding instruction with ethical conversations that encourage students to think about the greater impact of coding on others.

Future Studies

There is still work to be done in this area. One possible reason for a dearth of literature on ethics instruction in coding at the K-12 level is that incorporating ethics may be challenging due to its abstract nature. There is not always a clear correct answer. Instead, students must wrestle with core human values and non-ethical values which can often result in difficult classroom conversations. The authors hope this article encourages additional work, in both research and teaching, related to the integration of ethics and the impact of computing, and its role in the school library.

Outside of the school library, future research could explore habits of mind that are taught in computer science courses and the impact of ethical decision-making when integrated into K-12 coding classes. In addition, lesson plans that incorporate coding tutorials with ethical practice should be more readily available. Ethical case studies that are applicable to the K-12 environment could be developed. Specific examples for teaching the impact of computing should also be made available for teachers. Finally, school librarian and teacher education programs should include ethical frameworks. Webb (2019) recommends that universities encourage students studying computer science to obtain a dual degree "in computer science and political science, philosophy, anthropology, international relations, economics, creative arts, theology, and sociology" (p. 258). Additionally, Webb suggests that ethics be integrated into many courses instead of being taken as a standalone course that does not directly relate to computer science curriculum. While Webb recommends these changes for universities, similar cross-disciplinary discussions should be encouraged in both K-12 curriculum and information studies programs. Introducing students to these important topics earlier in their schooling will help young students to critically consider how computing can impact, sustain, or improve social inequities, as well as protect or infringe on people's rights.

Compliance with Ethical Standards

The authors declare that they have no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors.

References

2019 State of Computer Science Education: Equity and diversity. (2019). Retrieved from https://advocacy.code.org/2019_state_of_cs.pdf

Association for Computing Machinery. (2018). *ACM code of ethics and professional conduct.* https://www.acm.org/code-of-ethics

American Association of School Librarians. (2018). *AASL standards framework for learners* [PDF File]. Retrieved from https://standards.aasl.org/wp-content/uploads/2017/11/AASL-Standards-Framework-for-Learners-pamphlet.pdf

American Association of School Librarians. (2019). *National school library standards crosswalk with Code with Google's CS First Curriculum* [PDF File]. Retrieved from https://standards.aasl.org/wp-content/uploads/2019/11/aasl-standards-crosswalk-cs-first.pdf

Bruneau, E. G., Cikara, M., & Saxe, R. (2017). Parochial empathy predicts reduced altruism and the endorsement of passive harm. *Social Psychological and Personality Science*, *8*(8), 934–942. https://doi.org/10.1177/1948550617693064

Bynum, T. W. (2001). Computer ethics: Its birth and its future. *Ethics and Information Technology, 3*(2), 109-112. https://doi.org/10.1023/A:1011893925319

Bynum, T. W. (2017). Ethical challenges to citizens of 'The automatic Age': Norbert Wiener on the information society. In J. Weckert (Ed.), *Computer Ethics* (pp. 3-12). Routledge. https://doi.org/10.4324/9781315259697

Colby, J. (2015). 2,445 Hours of Code: What I learned from facilitating Hour of Code events in high school libraries. *Knowledge Quest, 43*(5), 12–17. https://files.eric.ed.gov/fulltext/EJ1063879.pdf

Cross, N. (2011). *Design thinking: Understanding how designers think and work*. Berg Publishers.

Dadlani, P., & Todd, R. J. (2015). Information technology and school libraries: A social justice perspective. *Library Trends, 64*(2), 329-359. https://doi.org/10.1353/lib.2015.0041

Davis, M. (2006). Integrating ethics into technical courses: Micro-insertion. *Science and Engineering Ethics, 12*(4), 717-730. https://doi.org/10.1007/s11948-006-0066-z

Duncan, C., & Bell, T. (2015, November). A pilot computer science and programming course for primary school students. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 39-48). ACM. https://doi.org/10.1145/2818314.2818328

Duncan, C., Bell, T., & Tanimoto, S. (2014). *Should your 8-year-old learn coding?* Proceedings of the 9th Workshop in Primary and Secondary Computing Education on – WiPSCE '14. https://doi.org/10.1145/2670757.2670774

Eubanks, V. (2018). *Automating inequality: How high-tech tools profile, police and punish the poor.* St. Martin's Press.

Falkner, K., & Vivian, R. (2015). Coding across the curriculum: Resource review. *Computer Science Education Research Group*. https://doi.org/10.13140/RG.2.1.2293.1929

Fiesler, C., Garrett, N., & Beard, N. (2020, February). What do we teach when we teach tech ethics? A Syllabi Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 289-295). https://doi.org/10.1145/3328778.3366825

Google for Education. (n.d.). *About CS First - CS First*. Google. https://csfirst.withgoogle.com/s/en/about.

Harrop, W. (2018). *Coding for children and young adults in libraries: A practical guide for librarians.* Rowman & Littlefield Publishers.

Hern, A. (2016). 'Partnership on AI' formed by Google, Facebook, Amazon, IBM, and Microsoft. *The Guardian*. https://www.theguardian.com/technology/2016/sep/28/google-facebook-amazon-ibm-microsoft-partnership-on-ai-tech-firms

Howard, A., & Borenstein, J. (2018). The ugly truth about ourselves and our robot creations: the problem of bias and social inequity. *Science and engineering ethics*, *24*(5), 1521-1536. https://doi.org/10.1007/s11948-017-9975-2

Hutchison, A., Nadolny, L., & Estapa, A. (2016). Using coding apps to support literacy instruction and develop coding literacy. *Reading Teacher, 69*(5), 493-503. https://doi.org/10.1002/trtr.1440

Google Inc. & Gallup Inc. (2016). Trends in the state of computer science in U.S. K-12 schools. http://goo.gl/j291E0

Johnson, G. M. (2020). Algorithmic bias: on the implicit biases of social technology. *Synthese*, 1-21. https://doi.org/10.1007/s11229-020-02696-y

Jones, S. (2016). Doing the right thing: Computer ethics pedagogy revisited. *Journal of Information, Communication & Ethics in Society, 14*(1), 33-48. https://doi.org/10.1108/JICES-07-2014-0033

K-12 Computer Science Framework. (n.d.). *A vision for computer science*. https://k12cs.org/a-vision-for-k-12-computer-science/

Kafai, Y., Burke, Q., & Resnik, M. (2014). *Connected code: Why children need to learn programming*. The MIT Press.

Kalogiannakis, M., & Papadakis, S. (2017, August). Pre-service kindergarten teachers acceptance of "ScratchJr" as a tool for learning and teaching computational thinking and Science education. In proceedings of the *12th Conference of the European Science Education Research Association (ESERA)*. https://www.ase.org.uk/system/files/Kalogiannakis%20et%20al_0.pdf

Karsenti, T. (2019). 12 reasons to learn coding at school. *EdCan Network*. https://www.edcan.ca/articles/coding-in-schools/

Kiang, D. (2015). Coding by design: A design-first approach. *Edutopia*. https://www.edutopia.org/blog/coding-by-design-first-approach-douglas-kiang

Lamb, A. (2016, 02). Makerspaces and the school library, part 2: Collaborations and connections. *Teacher Librarian, 43*, 56-60. https://www.thefreelibrary.com/Makerspaces+and+the+school+library%2C+Part+2%3A+collaborations+and...-a0445116713

Levin, S. (2019). 'Bias deep inside the code': The problem with AI ethics in Silicon Valley. *The Guardian*. https://www.theguardian.com/technology/2019/mar/28/big-tech-ai-ethics-boards-prejudice

Maner, W. (2002). Heuristic methods for computer ethics. *Metaphilosophy*, *33*(3), 339–365. https://doi.org/10.1111/1467-9973.00231

Martin, C. (2017). Libraries as facilitators of coding for all. *Knowledge Quest, 45*(3), 46-53. https://files.eric.ed.gov/fulltext/EJ1125376.pdf

Moor, J. H. (1999). Just consequentialism and computing. *Ethics and Information Technology, 1*(1), 61-69. https://doi.org/10.1023/A:1010078828842

Moorefield-Lang, H. (2015). Change in the making: Makerspaces and the ever-changing landscape of libraries. *TechTrends: Linking Research & Practice to Improve Learning, 59*(3), 107–112. https://doi.org/10.1007/s11528-015-0860-z

Moura, K. (2018). 5 ways to get your students coding across the curriculum [blog post]. https://www.kidsdiscover.com/teacherresources/5-ways-to-get-your-students-coding-across-the-curriculum/

Naidoo, J. C., & Sweeney, M. E. (2015). Educating for social justice: Perspectives from library and information science and collaboration with K-12 social studies educators. *Journal of International Social Studies*, *5*(1), 196-207. https://www.iajiss.org/index.php/iajiss/article/view/193/182

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. New York University Press.

Noble, S. U. (2013). Google search: Hyper-visibility as a means of rendering black women and girls invisible. *InVisible Culture*, (19). https://ivc.lib.rochester.edu/google-search-hyper-visibility-as-a-means-of-rendering-black-women-and-girls-invisible/

Oltmann, S. M. (2018). Ethics, Values, and intellectual freedom in school libraries. *School Libraries Worldwide*, *24*(1). http://dx.doi.org/10.14265.24.1.005

Osoba, O. A., & Welser IV, W. (2017). *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation. https://www.rand.org/pubs/research_reports/RR1744.html

Pasquale, F. (2016). *The black box society: The secret algorithms that control money and information.* Harvard University Press.

School Library Journal. (2019). School technology survey: U.S. School Libraries 2019. https://www.slj.com/?detailStory=School-Librarians-Excel-in-Coding-Instruction-Maker-Ed-Tech-SLJ-2019-Technology-Survey

Sherif, M., Harvey, O. J., White, B. J., Hood, W. R., & Sherif, C. W. (1961). *Intergroup conflict and cooperation: The Robbers Cave experiment*. University of Oklahoma.

Subramaniam, M., Kodama, C., Baylen, D. M., Burton, M., Fabicon, J. K., Hincks, K., Moniz, R., Smith, D., & Visser, M. (2019). *Computational thinking in libraries: Case studies of youth programs in action.* The American Library Association's Office for Information Technology Policy. https://www.ala.org/advocacy/sites/ala.org.advocacy/files/content/tools/RtC%20Case%20Studies_FINAL2.pdf

Smith, D. L. (2018). Coding for success. *Teacher Librarian, 45*(5), 13-16.

Uzunboylu, H., Kınık, E., & Kanbul, S. (2017). An analysis of countries which have integrated coding into their curricula and the content analysis of academic studies on coding training in Turkey. *TEM Journal*, *6*(4), 783-791. https://www.temjournal.com/content/64/TemJournalNovember2017_783_791.pdf

Villasenor, J. (2019). Artificial intelligence and bias: Four key challenges. *Brookings*. https://www.brookings.edu/blog/techtank/2019/01/03/artificial-intelligence-and-bias-four-key-challenges/

Visser, M. (2018). Ready to code with feathered friends. *ALA District Dispatch*. https://www.ala.org/tools/readytocode/resource/ready-code-feathered-friends

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when?. *Education and Information Technologies*, *22*(2), 445-468. https://doi.org/10.1007/s10639-016-9493-x

Webb, A. (2019). *The big nine: How the tech titans & their thinking machines could warp humanity.* Public Affairs.

Wiener, N. (1954). *The Human use of human beings: Cybernetics and society.* Houghton Mifflin.

Willett, R. (2016). Making, makers, and makerspaces: A discourse analysis of professional journal articles and blog posts about makerspaces in public libraries. *Library Quarterly, 86*(3), 313–329. https://doi.org/10.1086/686676

Wine, L. D. (2016). School librarians as technology leaders: An evolution in practice. *Journal of Education for Library and Information Science, 57*(2), 207-220. https://doi.org/10.12783/issn.2328-2967/57/2/12