

Week14

“What is literacy and what does it mean to call computer programming literacy? ”

Literacy traditionally refers to the ability to read and write in a particular language. However, the concept has expanded to encompass other forms of communication, such as digital literacy, which includes the ability to use and understand digital technology. Computer programming literacy, therefore, refers to the ability to read and write computer code, and to use it to create, modify, and understand software.

Calling computer programming a literacy is a way of acknowledging its importance in today's society. In a world that is increasingly reliant on technology, being able to understand and create software is becoming a necessary skill. In many ways, it is similar to the way that reading and writing literacy was necessary in the past. Just as the ability to read and write allowed people to participate in society and access information, computer programming literacy allows people to participate in the digital world and to create new solutions to problems.

Furthermore, computer programming literacy is becoming increasingly important as the field of technology continues to grow and evolve. The ability to code is becoming a valuable skill in many different industries, from finance to healthcare to education. In addition, being able to code allows people to automate repetitive tasks and to create more efficient workflows. This can lead to increased productivity, creativity, and innovation.

In conclusion, computer programming literacy is a vital skill in today's world. It allows people to participate in the digital world, create new solutions to problems, and automate repetitive tasks. As the importance of technology continues to grow, so does the importance of computer programming literacy.

“New ideas to help address the concept of coding literacy”

Reading the assigned texts made me think about the notion of coding literacy in new ways. One of the ideas that stood out to me was the notion of viewing programming from the perspective of literacy, and literacy from the perspective of programming. Vee argues that this shift in perspective helps us understand programming beyond a technical level, and in its historical, social, and conceptual contexts. This made me think about how our current understanding of literacy and programming is often limited to technical skills, and how we need to broaden our understanding to encompass the social and cultural contexts in which they are embedded.

Similarly, the introduction chapter of Live Coding: A User's Manual highlights the performative and creative aspects of live coding, and how it can be used as a tool to

explore deeper questions about contemporary cultural production and computational culture. This made me think about the ways in which coding literacy can be used to foster creativity and expression, rather than just technical proficiency. As the authors note, "Live coding is about how people interact with the world and each other via code." This reminded me that programming is not just a solitary activity, but one that is inherently social and collaborative.

The video of Sam Aaron discussing programming as performance further reinforced the idea of programming as a creative and expressive activity. He argues that programming is not just about solving technical problems, but about exploring new possibilities and pushing boundaries. He also emphasizes the importance of celebrating creativity within all aspects of programming. This made me think about how we can encourage and support more diverse voices and perspectives in programming, and how we can use programming as a tool for social change and innovation.

Overall, these readings helped problematize the notion of coding literacy as just a technical skill, and highlighted the importance of understanding programming in its social, cultural, and creative contexts. I was particularly inspired by the ideas of programming as performance, and the potential of live coding to open up deeper questions about contemporary cultural production and computational culture.