# **Organizing Data Discussion Guide (for use during or after reading)**

1. What are data structures and why are they important? (Data Is Information, What Is a Data Structure?, p. 4-7)
   1. Data structures are the organization of data by a computer program. They are important because computers use a lot of data and need an organized way to store it all.
2. Think about the unicorn examples on pages 8-11. Explain how a data structure is helpful when coding with many variables and data values. (What Is a Data Structure?, p. 6-11)
   1. Programmers often need to create codes that include many variables and data values. They use data structures in order to keep their codes simple, organized, and efficient. For example, programmers can use a structure to combine many variables into one to make the coding process easier, like in the example on pages 8-9. Instead of needing to keep track of many individual variables, programmers can use structures to keep all the details of a program organized and easy to manipulate.

1. What is an array? How does it store and organize data? (Arrays, p. 12-13)
   1. An array is a specific type of data structure that stores a sequence of related information. For example, on pages 12 and 13 we see how an array can be used to combine all the information about each unicorn into one group.
2. When thinking about data structures, what does the word “element” mean? (Arrays, p. 12-13)
   1. When thinking about data structures, the word “element” refers to the smallest part of a data structure. Each element can store one piece of information. In the example on pages 12 and 13, each element represents one unicorn. Because each programmed unicorn also included its own data structure, the array not only includes each of the 5 unicorns, but also all the variables they contain.
3. Describe a stack data structure. Why do some programmers refer to it as “Last In, First Out”? (Stacks, p. 14-15)
   1. A stack data structure is one in which data can only be removed from a list in the reverse order that it was added. It can be helpful to think about a physical stack of items where the only way to remove one is directly from the top of the stack. Stack data structures are sometimes referred to as “Last In, First Out” because the last element added to a stack is the first one to be removed.
4. How do web browsers use stacks to help people navigate? (Stacks, p. 16-17)
   1. Web browsers make use of stack structure to help people navigate backwards (or forwards!) to web pages they have visited. Web browser programs put browsing history into a stack. When we click the back button on a web browser, we are removing the top element from the stack, sending us “backwards” one web page.
5. Describe a queue data structure. Use an example to show when queues are beneficial to programmers. (Queues, p. 18-21)
   1. A queue data structure is one in which data is removed from a list in the same order that it was added. It can be helpful to think about a line (sometimes referred to as a queue). The first element in the line is the first one to be removed. Likewise, new elements are added to the back of the line. Queue structure can be helpful when something needs to be done in a certain order, such as printing the pages of a book.
6. What is set structure? Explain why a playlist is a good example of set structure. (Sets, p. 22-25)
   1. A set data structure is one that stores all the data in one variable. It can be easy to add or remove data from set structure. A playlist of songs or videos is a good example of a set data structure because it shows how we can easily add or remove elements from a set, just as we do with songs and videos in a playlist.
7. Describe a hash data structure and when might it be useful to use. (Hashes, p. 26-27)
   1. A hash data structure is one that links each element to a matching piece of data. Hash structures can be useful when programmers need to connect two groups of data to one another.
8. Why is it beneficial for computer programmers to know multiple ways to organize and store data? (Hashes, p. 28-29)
   1. It is beneficial for computer programmers to know many ways to organize and store data because different data structures are useful for different purposes. Programmers must understand their goal and choose the best structure for it.