# **Order in Coding Discussion Guide (for use during or after reading)**

1. What is control flow and why is it important for programmers to keep in mind when coding? (Order!, p. 4-5)
   1. Control flow is the order in which a computer follows the steps of a computer program. It is important for programmers to keep control flow in mind when coding because programs will not produce the intended outcome if they are out of order or otherwise unorganized.
2. How can loops and conditions support the work of control flow? (Control Flow, p. 6-9)
   1. Elements like loops and conditions can be used to support the work of control flow. Adding these elements tells the computer when to carry our specific program instructions and allows the programmer to be in more control of the flow of their program.
3. What is sequencing control flow? How does it work and when should it be used? (Sequencing, p. 10-11)
   1. Sequencing is a control flow in which the computer program follows the steps of code in the order they are written. Sequencing control flow works well for many straightforward computer programs, such as those used to add one new piece of information at a time.
4. What is selection control flow? How does it work and when should it be used? (Selection, p. 12-13)
   1. Selection is a control flow in which specific lines of code are skipped over unless certain conditions are true. Selection control flow works well when a new outcome is needed only for a certain situation or condition.
5. What is iteration control flow? How does it work and when should it be used? (Iteration, p. 14-15)
   1. Iteration is a control flow in which certain lines of code are repeated. Iteration control flow works well when a series of steps must be repeated over and over again until the task is finished.
6. What can programmers do to ensure the data provided is useable? (Data Sorting, p. 16-19)
   1. In order to ensure data is in a useful order, programmers can organize or sort their data by certain attributes. Frequently, programmers will use data sorting algorithms to make sure their data is organized in a useful way.
7. Explain how an insertion sort can be used to order data. (Insertion Sort, p. 20-21)
   1. An insertion sort is a sorting algorithm in which the elements in a list are sorted one by one. Here, each element is taken out of the list, considered, and inserted back into the correct place. This is a common data sorting algorithm, but it might not be the most efficient for all situations.
8. How is a selection sort similar to and different from an insertion sort? (Selection Sort, p. 22-23)
   1. A selection sort is a sorting algorithm that goes through the elements in a list one by one, choosing the minimum or maximum element. Selection sorts are similar to insertion sorts because they both consider elements one at a time. However, a selection sort compares data by choosing each element by its minimum or maximum effect. For example, in order to sort giraffes by height, a selection sort would start with the tallest giraffe, then the next tallest, then the next tallest, and so on until all the giraffes are sorted.
9. What is a merge sort? Why is it considered to be more useful than other types of sorts? (Merge Sort, p. 24-27)
   1. A merge sort is a sorting algorithm that splits up the elements in a list into groups and pairs before recombining them in the correct order. Merge sorts are more complex than selection or insertion sorts but can be more useful and more efficient, depending on the situation. This is because merge sorts involve splitting the data into smaller and smaller groups until they cannot be split apart any further. Then, elements are compared and recombined in the correct order.
10. Why is it important to have a variety of ways to sort and organize data? (Data Sorting, Insertion Sort, Selection Sort, Merge Sort, p. 16-29)
    1. It is important to have a variety of ways to sort and organize data because different programs require different sorting methods. In addition to insertion, selection, and merge sorts, other sorting algorithms exist, such as the divide-and-conquer sorting method. Programmers should know a variety of ways to sort data so they can choose the best method for their intended purpose.