# **Organising Data Comprehension Check**

For questions 1-4, match each vocabulary term to the correct definition:

|  |  |
| --- | --- |
| 1. Data | a. the smallest part of a data structure that can store one piece of information |
| 2. Data Value | b. a value, or piece of information, that can change |
| 3. Element | c. information that a computer processes or stores |
| 4. Variable | d. a number or other piece of information that is used in a computer program |

1. Data –
2. Data Value –
3. Element –
4. Variable –
5. What is an array? How does it store and organise data?
6. Describe a stack data structure. Why do some programmers refer to it as “Last In, First Out”?
7. Describe a queue data structure. Use an example to show when queues are beneficial to programmers.
8. What is set structure? Explain why a playlist is a good example of set structure.
9. Describe a hash data structure and when might it be useful to use.
10. Why is it beneficial for computer programmers to know multiple ways to organise and store data?

# **Organising Data Comprehension Check Answer Key**

For questions 1-4, match each vocabulary term to the correct definition:

|  |  |
| --- | --- |
| 1. Data | a. the smallest part of a data structure that can store one piece of information |
| 2. Data Value | b. a value, or piece of information, that can change |
| 3. Element | c. information that a computer processes or stores |
| 4. Variable | d. a number or other piece of information that is used in a computer program |

1. Data – c
2. Data Value – d
3. Element – a
4. Variable – b
5. What is an array? How does it store and organise data?
   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.
6. Describe a stack data structure. Why do some programmers refer to it as “Last In, First Out”?
   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.
7. Describe a queue data structure. Use an example to show when queues are beneficial to programmers.
   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.
8. What is set structure? Explain why a playlist is a good example of set structure.
   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.
9. Describe a hash data structure and when might it be useful to use.
   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.
10. Why is it beneficial for computer programmers to know multiple ways to organise and store data?
    1. It is beneficial for computer programmers to know many ways to organise and store data because different data structures are useful for different purposes. Programmers must understand their goal and choose the best structure for it.