

1.1 Arrays in Data Structure | Declaration, Initialization, Memory representation

Jenny's Lectures CS IT

be stored in memory now as you can see theory is stored in 20 bytes memory starting from the 0th byte and ending at the 19th byte. So theory is stored in consecutive bytes in memory. Now If I want to access any of these elements, I will use the indexing operator, which is the square bracket []. So I can say here [0] that is the first element of this array. Now if I want to access any other element. I will use the indexing operator again and this time I will use the plus sign (+). So I can say here [1] that is the second element of this array and so on until I reach the last element of this array, which is [19]. So this is how you can access any of these elements of the array using the indexing operator [] and then using the plus sign (+) see now Another way to initialize an array at runtime is by using a pointer variable. See now suppose we have a pointer variable called p that points to an area in memory that contains an array called [UNK] fine now at runtime. We can say p->elements[0] that is referencing the first element of [UNK] See now Another way to initialize an array at runtime is by using a const pointer variable. See now suppose we have a const pointer variable called p that points to an area in memory that contains an array called [UNK] fine now at runtime. We can say const p->elements[0] that is referencing the first element of [UNK]. See now Another way to initialize an array at runtime is. a dynamic pointer variable see now suppose we have a dynamic pointer variable called P that points to an area in memory that contains an array called [UNK] fine. Now at runtime, we can say p->elements[0] that is referencing the first element of [UNK], but what if we want to access more than one element from this array at runtime. What would we do see now suppose we want to access both [1] and [2] from this array at runtime. We can do this by using the two indexing operators [] and (+) Like this see now another way to access more than one element from this array at runtime is by using a compound assignment operator (=). Like this see now suppose we want to access both [1] and [2] from this array, but we also want to assign them both to another variable called V right so we do like this P = V[1] + V[2] see now one more way to access more than one element from this array at runtime is The data for this array is stored in 4 bytes starting at offset 0. The base address of this array is 100. The address of the first element in this array is 0. The author suggests that in order to access an array's data, you need to know the array's base address and the size of the data type being used. You can then use the base address plus the data type's size to calculate the array's actual address.

always be allocated consecutively in memory size of data. Items in an array is not restricted by the amount of memory available on the machine on which the program is running. Small arrays may be allocated in memory. Even if the program is not running efficiently. Large arrays may be allocated only if there is enough memory available on the machine. This passage provides insights into the [UNK] [UNK] data-type of an array [UNK] advantages and disadvantages of the array [UNK] to access elements in an array..