Dyln Xiong

CSI 2025

9/25/2025

So for this writing portion of the homework for Activity_4_1, I will just be looking at my codespace on GitHub and try to describe what I am seeing and reading. As I describe them, I will also try and record my thoughts on what I think things mean and stuff like that. What is an array? I have no clue on what an array is at the moment because I'm pretty sure I forgot. I know that an array can hold elements of integers, characters, strings of characters, and etc. The structure of an array code would look like this, "std::array< (element type) , (size) > (name of array) {size of your array with its elements}". An example of this is "std::array<int, 5> A {5, 10, 15, 20, 25};". An array has a certain size that depends on what you have inside your point brackets in your array. Also, you must have a ";" at the end of each of your codes because it stands for and meaning you are continuing onto your next code. I'm pretty sure that's what it means. You can print these elements out by using a code known as "std::printf();". To print an array of integers as known as int, you can write "std::printf("%d\n", (name).at(any number)); and this will give you an integer from your array from the number you input to find at. But to use any of these codes you must have "#include <cstdio>" and "#include<array>". These will let your computer know that you include the libraries of these types of code which allows you to use the codes from those libraries. The names after "#include" are known as headers and they are what labels the libraries of things such as "<array>" and "<cstdio>" I think. The type of integers that come out of your print is different depending on what types of arrays you are doing. The example I used was an integer array which means numbers. Now I don't know if the size of an array is the maximum number of elements that can be printed out but I do think it isn't. Also, I don't know if you can print any of these arrays differently from how I showed like "("%d\n", (name).(number)).

Moving on, I will now be trying to describe the things I see in vectors and be trying to record my thoughts on them. Now, what is a vector? To be honest, I also forgot the term for what a vector is. All I know from seeing it on my code space is that it is much more simpler to use than an array is. A vector doesn't need size like an array does to function. It also uses elements of similar types like an array. Elements such as integers, characters, strings of characters, and etc. The structure of a vector would look like this, "std::vector<(element type)> (name of vector) = {elements with however much you want in it}". Also, to be able to use a vector at all, you must have "#include<vector>", this lets the computer know that you are using the library of vectors. The library of vectors contains the codes for a vector I think. Now an example of a vector would be, "std::vector<int> A = {1, 2, 3, 4, 5}". You can have however many elements you want with a vector as it has no size limit you need to stop at or reach. Also, to print a vector, it would be exactly the same way you would print an array. Also, to use elements of strings of characters, you must have “#include<string>”. Also, when the characters and strings of char are written, they must have ‘(for char) ‘ and “(for string of char) “ to work. Now I have no clue as to why vectors work differently from arrays but all I know is that they are more simple than arrays when it comes to writing them.