# Lab Work 4

**Learning Objectives**

* Arrays holding multiple values
* Accessing array elements
* No bound checking in C++
* Array initialization
* Processing array contents
* Arrays as function arguments

**Lab exercise 1**

Unzip all codes from Week4 Example codes and observe the outcome of each code.

**Lab Exercise 2**

Please answer all the short questions below:

1. Define the following arrays: (only definition)
   1. empNums, a 100 element array of int

**int empNum[100];**

* 1. payRates, a 25 element array of floats

**float payRates[25];**

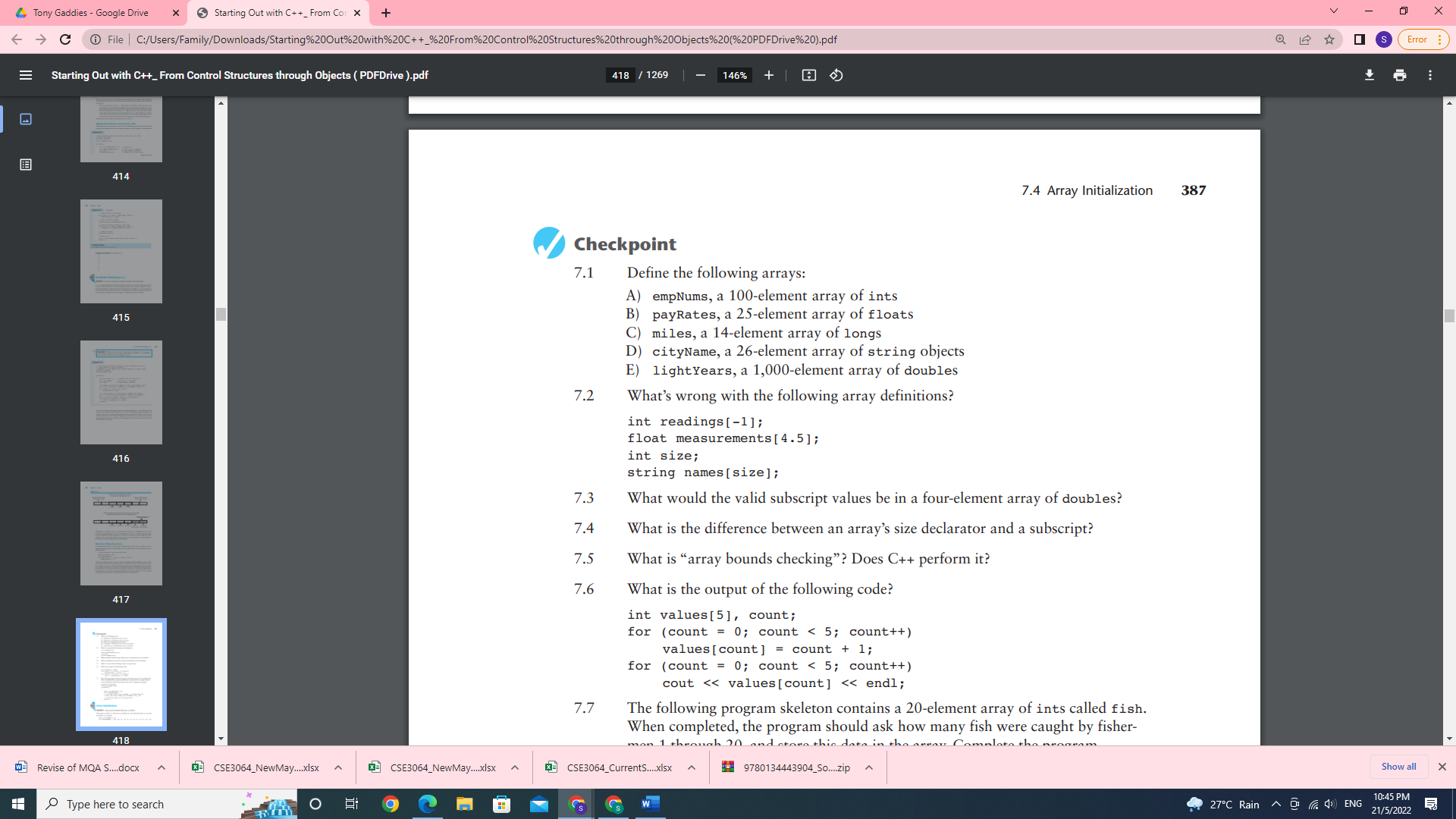
* 1. miles, 14 element array of longs

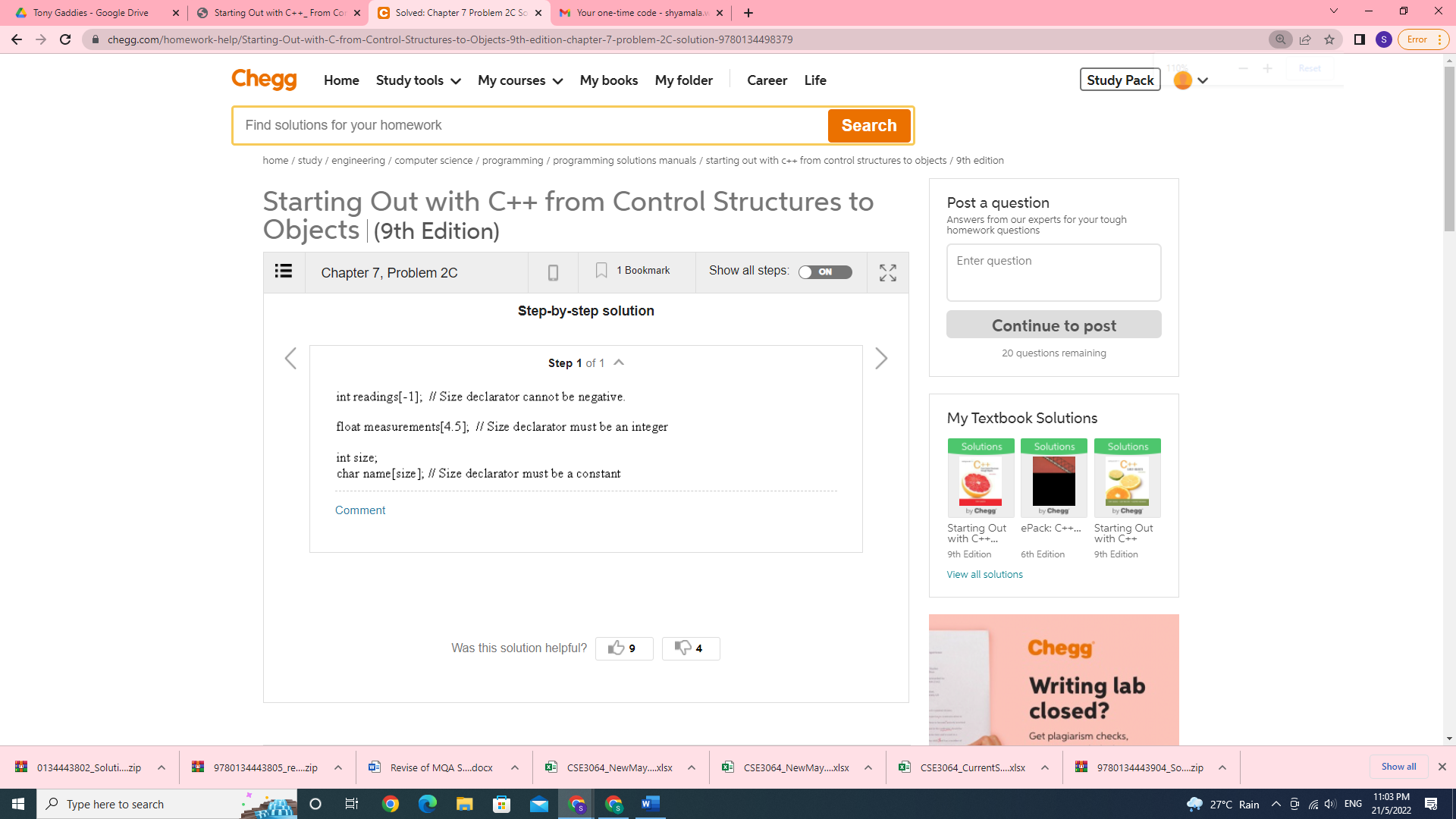
**long miles[14];**

* 1. cityName, 26 element array of string objects

**string cityName[26];**

1. What is wrong with the following array definitions?





1. What would the valid subscript values be in a four-element array of double called s?

**The valid subscript value would be 0,1,2,3.**

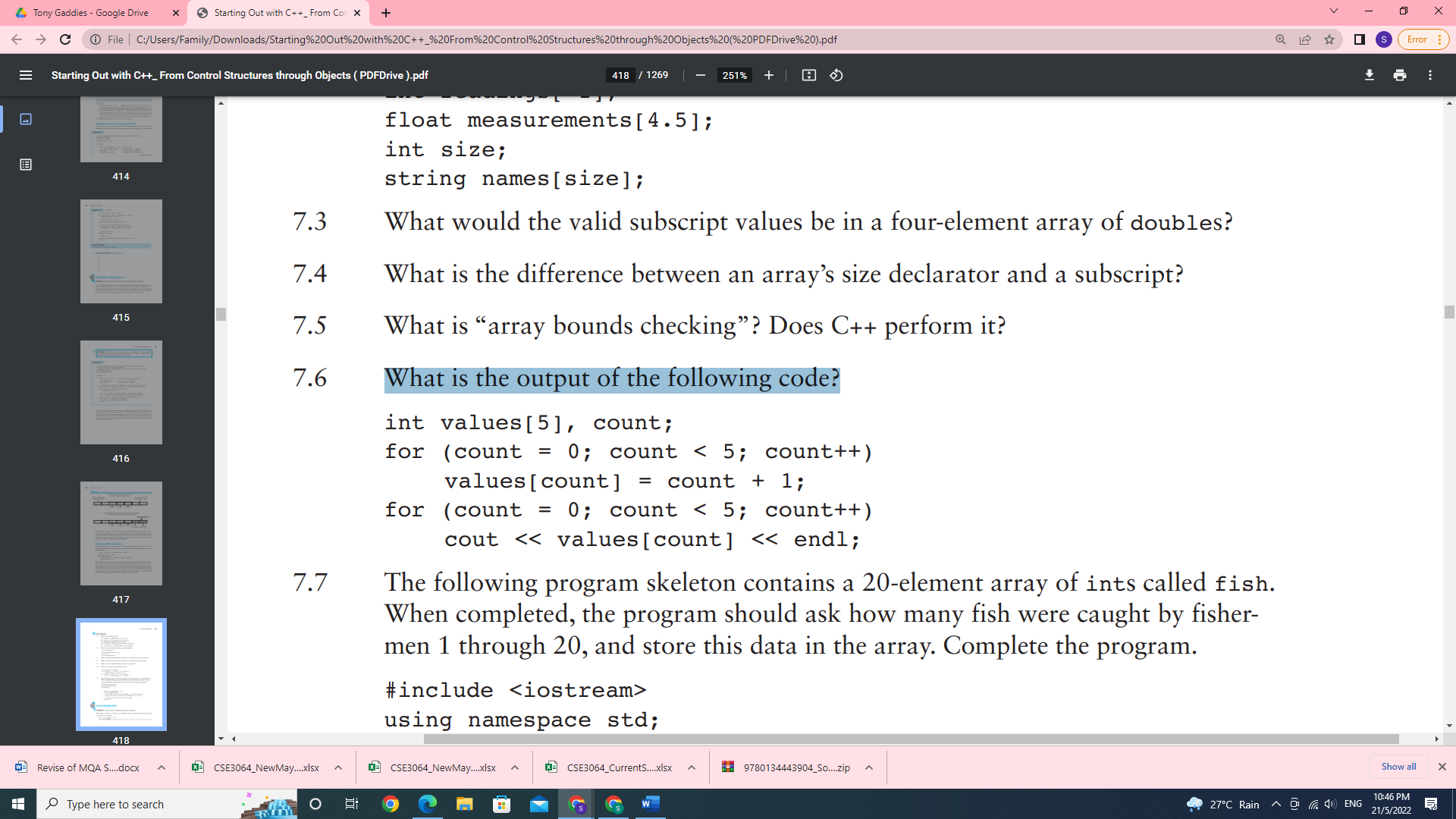
1. What is the difference between an array’s size declarator and a subscript?

**The number inside the brackets of an array definition is the size declarator. The size declarator is used in array declaration statement. It specifies the number of elements in the array. The number inside the brackets of an assignment statement or any statement that works with the contents of an array is a subscript. A subscript is used to access an individual element in an array.**

1. What is “array bounds checking”? Does C++ perform it?

**Array bounds checking is a safeguard provided by some languages. It prevents a program from using a subscript that is beyond the boundaries of an array. In C++ does not perform array bounds checking.**

1. What is the output of the following code?



**The code will display**

**1**

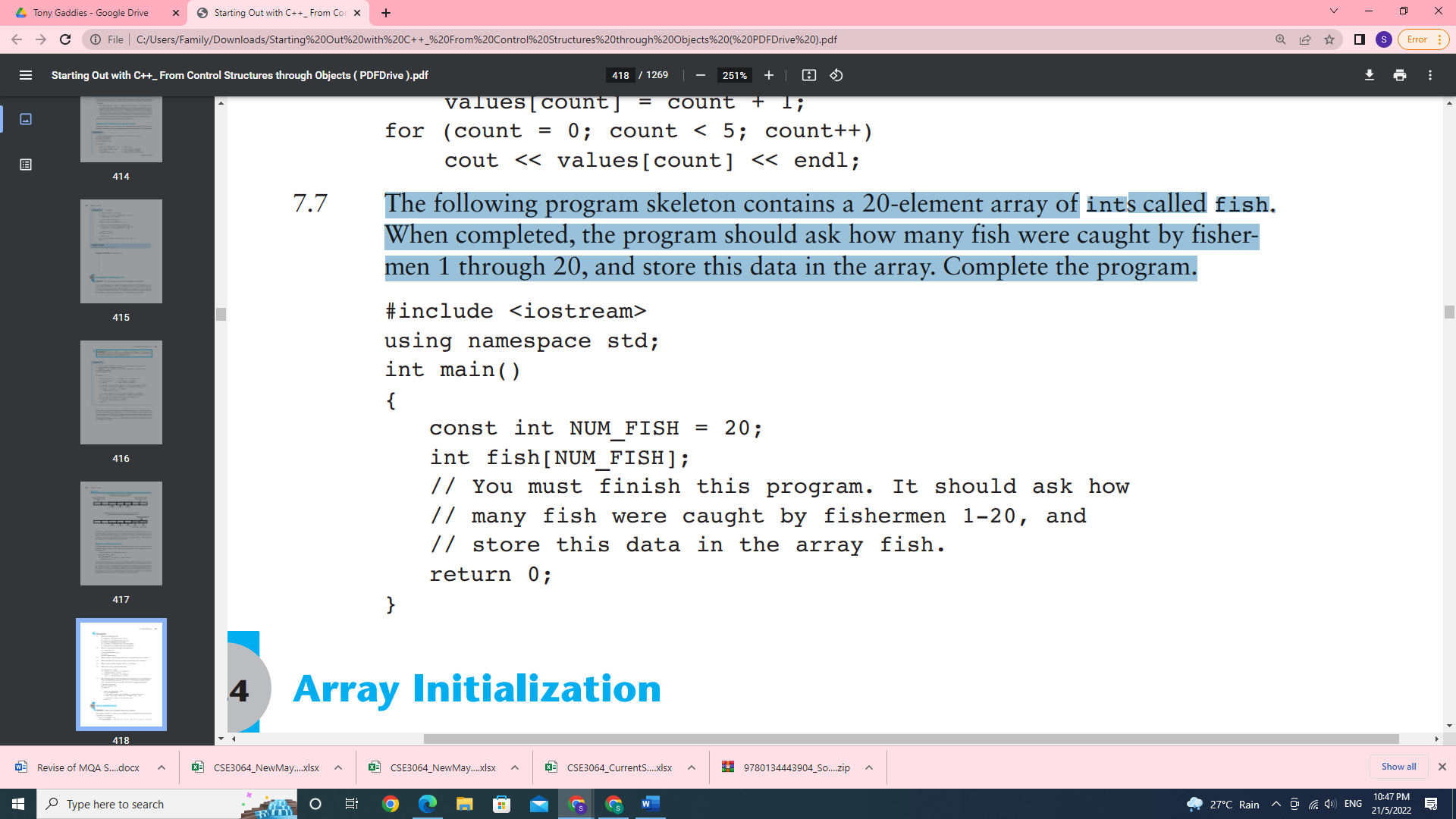
**2**

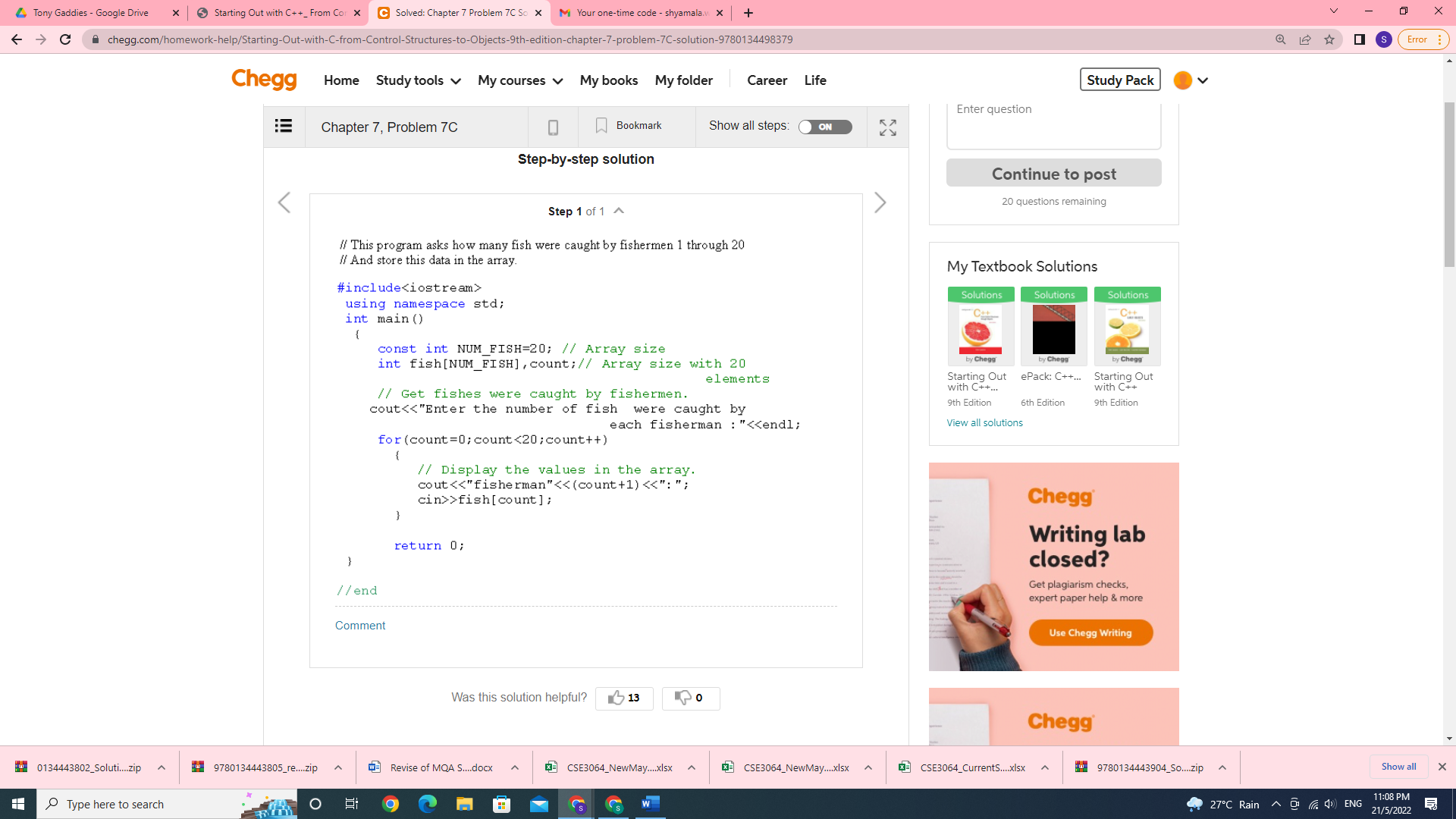
**3**

**4**

**5**

1. The following program skeleton contains a 20-element array of int s called fish. When completed, the program should ask how many fish were caught by fishermen 1 through 20, and store this data in the array. Complete the program.





1. Given the following array definitions

double array1[4] = {1.2, 3.2, 4.2, 5.2};

double array2[4];

will the following statement work? If not, why?

array2 = array1;

**Yes it will work all the values from array 1 will be copied to array 2. Both the array is the same type and same size**

1. When an array name is passed to a function, what is actually being passed?

**Depends on how it is passed, if it is passed by value then the entire array is copied and passed to the function. If it is passed by reference then it will pass the memory address only.**

**Example of passing by value**

**// C++ Program to display marks of 5 students**

**#include <iostream>**

**using namespace std;**

**// declare function to display marks**

**// take a 1d array as parameter**

**void display(int m[5]) {**

**cout << "Displaying marks: " << endl;**

**// display array elements**

**for (int i = 0; i < 5; ++i) {**

**cout << "Student " << i + 1 << ": " << m[i] << endl;**

**}**

**}**

**int main() {**

**// declare and initialize an array**

**int marks[5] = {88, 76, 90, 61, 69};**

**// call display function**

**// pass array as argument**

**display(marks);**

**return 0;**

**}**

**Passing by reference**

**Void printArray(int \*a,int size)**

**This means you are only sending the address of the array to the function. The same array is being updated by the function. This we shall learn indepth when we do pointers**

1. When used as function arguments, are arrays passed by value?

**Can be passed as value or reference. Same question like question 9**

1. The following program skeleton, when completed, will ask the user to enter 10 integers, which are stored in an array. The function avgArray, which you must write, is to calculate and return the average of the numbers entered.

#include using namespace std;

// Write your function prototype here

int main()

{ const int SIZE = 10;

int userNums[SIZE];

cout << "Enter 10 numbers: ";

for (int count = 0; count < SIZE; count++)

{

cout << "#" << (count + 1) << " ";

cin >> userNums[count];

}

cout << "The average of those numbers is ";

cout << avgArray(userNums, SIZE) << endl;

return 0;

}

//

//

Write the function avgArray here.

//

**#include<iostream>**

**using namespace std;**

**double avgArray(int arr[],int n){**

**double total = 0;**

**for(int i=0;i<n;i++)**

**total += arr[i];**

**return total/n;**

**}**

**int main(){**

**const int SIZE=10;**

**int userNums[SIZE];**

**cout<<"Enter 10 numbers:";**

**for(int count = 0;count<SIZE;count++){**

**cout<<"#"<<(count+1)<<" ";**

**cin>>userNums[count];**

**}**

**cout<<"The average of those numbers is ";**

**cout<<avgArray(userNums,SIZE)<<endl;**

**return 0;**

**}**