

Seatwork 4.2

Pointers

Course Code: CPE 007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: September 18, 2025
Section: CPE11S1	Date Submitted: September 18, 2025
Name(s): Jaime Luis M. Demain	Instructor: Engr. Jimlord M. Quejado

6. Output

Code:

```
1 #include <iostream>
2 using namespace std;
3
4 ▷ int main() {
5     const int size = 10;
6     int scores[size] = {95,85,78,88,92,80,75,80,89,91};
7     for (int i = 0; i < size; i++) {
8         cout << scores[i] << " ";
9     }
10    cout << endl;
11    for (int i = 0; i < size; i++) {
12        cout << "address of elements " << i << ":" << &scores[i] << endl;
13    }
14    int *scorePtr;
15    scorePtr = &scores[9];
16    cout << *scorePtr << endl;
17    cout << scorePtr << endl;
18    int numBytes = sizeof(scores);
19    cout << "The number of bytes of this array is: " << numBytes << endl;
20    return 0;
21 }
```

Output:

```
"C:\Users\Jaime Luis\CLionProjects\untitled\cmake-build-debug\untitled.exe"
95 85 78 88 92 80 75 80 89 91
address of elements 0: 0x7b3a5ff670
address of elements 1: 0x7b3a5ff674
address of elements 2: 0x7b3a5ff678
address of elements 3: 0x7b3a5ff67c
address of elements 4: 0x7b3a5ff680
address of elements 5: 0x7b3a5ff684
address of elements 6: 0x7b3a5ff688
address of elements 7: 0x7b3a5ff68c
address of elements 8: 0x7b3a5ff690
address of elements 9: 0x7b3a5ff694
```

```
91
0x7b3a5ff694
The number of bytes of this array is: 40
Process finished with exit code 0
```

7. Supplementary Activity

What I did in this seatwork is to print out the address of every and separate elements, and to print out the amount of bytes of an array. The first step is that I initialize an array and print out every value of an array. Next is I will now print out the address of every values. To do this, I must utilize the "&" operator in order to print out the address of the value (elements). It will undergo the same process as printing out the values of an index when it comes to printing out the address of the value, the difference is that I will insert "&" before "scores[i]" language so that the program will learn that I wil print out the address of an array. And then, I will now proceed on printing out the address of separate values. To do this, I must declare a word first "scorePtr" that will be used to print out the address. What I did is that I declare a value first using "*scorePtr" wherein the "*" serves as a dereference operator wherein it will access the value stored at the memory address pointed to by a pointer that will be pointed out in the next line. After that, I will now assign "scorePtr" to a specific index that I would like. We chose the 9th index (a 10th value of an array). After this, I will now print out the value and its address. After this, I would now proceed on processing out the size of an array. To do this, I will utilize a "sizeof(array's name)" to process the total size (bytes) of an entire array. After processing out, I would now proceed on printing out the total size of an array with a message. That's a whole process of determining the address of array, utilizing pointers, and determining the size of an array.

8. Conclusion

What I learned in this activity is on how to utilize a pointer, how to find the address of a particular value of an array, and to determine the total size of an array. In order to do this, I need to learn new operators and languages particularly "*", "&", and "sizeof". Printing out the address of each value of an array is similar to printing out the value of an array, I just need to insert "&" before the name of an array. After that, I learned pointers so that it will print out the particular value of an array when you print out its address. You print out the pointers first for value before printing out the address. And then, I will now process out "sizeof" to print out the total size (byte) of an array. That will be the end of the conclusion.