

Activity No. 4.2

Seatwork 4.2: Pointers

Course Code: CPE007

Program: Computer Engineering

Course Title: Programming Logic and Design

Date Performed: Sep. 18, 2025

Section: CPE11S1

Date Submitted: Sep. 18, 2025

Name(s): Tobias, Lawrence C.

Instructor: Engr. Jimlord M. Quejado

6. Output

bleee.cpp

```
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
8      for (int i = 0; i < size; i++) {
9          cout << scores[i] << " ";
10     }
11
12     cout << endl << endl;
13     for (int i = 0; i < size; i++)
14         cout << "address of element " << i << ": " << &scores[i] << endl;
15
16     cout << endl << endl;
17     int *scorePtr;
18     scorePtr = &scores[0];
19
20     cout << "the address of the array[0]: " << scorePtr << endl;
21     cout << "the dereferenced pointer: " << *scorePtr << endl;
22     cout << endl << endl;
23
24     int numBytes = sizeof(scores);
25     cout << "The number of bytes of the array is: " << numBytes << endl;
26
27     return 0;
28 }
```

```
C:\Dev-Cpp\bleee.exe  X  +  v
95 85 78 88 92 80 75 80 89 91

address of element 0: 0x6ffdf0
address of element 1: 0x6ffdf4
address of element 2: 0x6ffdf8
address of element 3: 0x6ffdfc
address of element 4: 0x6ffe00
address of element 5: 0x6ffe04
address of element 6: 0x6ffe08
address of element 7: 0x6ffe0c
address of element 8: 0x6ffe10
address of element 9: 0x6ffe14

the address of the array[0]: 0x6ffdf0
the dereferenced pointer: 95

The number of bytes of the array is: 40

-----
Process exited after 2.058 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

The program creates an array of 10 numbers called scores, with 10 numbers in it. And the numbers have all already been assigned a value. The program prints out all the numbers in the array at first so we can see the contents. Then the program prints out the memory address of every element, which tells the memory address of every element and the values in the computer memory. Then the program creates a pointer called scorePtr that points to the first element for the pointer, which can access values on arrays. Finally, the sizeof operator is used to calculate the number of bytes the entire array occupies in memory. The program prints the result.

8. Conclusion

Through this activity I learned how arrays, memory, and pointers interact. It prints all the numbers in the array and also prints out the memory address of each number. The pointer function "scorePtr" allowed me to see how we can point to the first element and see both its address and element. I learned more about how arrays are stored in the computer and how pointers help us use arrays better. I think I performed well in this activity because I was able to understand the base elements of arrays and pointers, but I need to improve more to understand memory address and practice with pointers so I can master them.