Hands-on Activity 4.2	
Arrays	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:9/11/25**
Section:CPE11S1	**Date Submitted:9/13/25**
Name(s): Niel Vincent B. Condino	**Instructor: Engr. Jimlord M. Quejado**

6. Output

Code

```cpp
#include <iostream>
using namespace std;

int main() {
    int n[10];

    // Initialize array elements to 0
    for (int i = 0; i < 10; i++) {
        n[i] = 0;
    }

    cout << "Element   Value" << endl;

    // Print index and value
    for (int i = 0; i < 10; i++) {
        cout << "    " << i << "        " << n[i] << endl;
    }

    return 0;
}
```

The code first initializes an int array named “n” with a size of 10. It then initializes the array elements to zero using a for loop. It then prints “Element Value” with a line break at the end.After that, it prints the index and values using a for loop. Lastly, it returns 0.

Output

```
Element   Value
    0        0
    1        0
    2        0
    3        0
    4        0
    5        0
    6        0
    7        0
    8        0
    9        0


--------------------------------
Process exited after 0.009308 seconds with return value 0
Press any key to continue . . .
```

Code

```cpp
#include <iostream>
using namespace std;

int main() {
    int n[10] = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};

    cout << "Element  Value" << endl;

    for (int i = 0; i < 10; i++) {
        cout << "    " << i << "       " << n[i] << endl;
    }

    return 0;
}
```

First, it declares an int array named “n” with a size of 10 that has a value of 32,27,64,18,95,14,90,70,60, and 37.Then it prints “Element Value”. It then prints the element with its corresponding value. Lastly it returns zero.

Output

```
Element  Value
    0       32
    1       27
    2       64
    3       18
    4       95
    5       14
    6       90
    7       70
    8       60
    9       37


--------------------------------
Process exited after 0.01439 seconds with return value 0
Press any key to continue . . .
```

Code

```cpp
#include <iostream>
using namespace std;

#define SIZE 12

int main() {
    int a[SIZE] = {1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45};
    int total = 0;

    for (int i = 0; i < SIZE; i++) {
        total += a[i];
    }

    cout << "Total of array element values is " << total << endl;
    return 0;
}
```

The code first defines “SIZE” with the value 12. It then initializes an int array named “a” with the size of SIZE which is equal to 12 with the value of 1,3,5,4,7,2,99,16,45,67,89,and 45. It then initializes a variable named “total” with the value of zero. After that it loops through the array then adds each value together. After that, it prints “Total of array element value is “ concatenated with the value of total at the end and a line break. Lastly it returns 0.

Output

```
C:\Users\TIPQC\Downloads\d
Total of array element values is 383

--------------------------------
Process exited after 0.0116 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

1.Code:

```cpp
#include <iostream>

int main(){
    int arraySize = 10;

    int value[10] = {19, 3, 15, 7, 11, 9, 13, 5, 17,1};

    std::string histogram[arraySize];

    for (int j = 0; j < arraySize;j++){
        for (int h = 0; h < value[j];h++ ){
            histogram[j] = histogram[j] + "*";
        }
    }

    std::cout << "Element\tValue\tHistogram"<<std::endl;

    for (int p = 0; p < arraySize; p++){
        std::cout << p << "\t" << value[p] << "\t" << histogram[p]<<std::endl;
    }

    return 0;
}
```

First, it declares arraySize to have a value of 10. After that it declares the array named “value” with a size of 10 with the values 19,3,15,7,11,9,13,5,17,1. After that it declares the string array named “histogram” with a size of the value of a variable named “arraySize”. Next, it uses a “for” loop that loops up to the value of the arraySize variable but inside this loop is another loop with a function of counting the value of the corresponding value inside the “value” array and then adding the corresponding number of “*” at that index on the “histogram” array the same index as in the “value” array . After that it prints ”Element Value Histogram” with a line break at the end. Using a for loop it prints the element index, value and histogram accordingly with a line break at the end. Lastly it returns 0.

Output:

```
Element Value   Histogram
0       19      ******************
1       3       ***
2       15      ***************
3       7       *******
4       11      ***********
5       9       *********
6       13      *************
7       5       *****
8       17      *****************
9       1       *

--------------------------------
Process exited after 0.01224 seconds with return value 0
Press any key to continue . . .
```

2.Code:

```cpp
#include <iostream>

int main(){

    int responses[40] = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10, 1, 6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7, 5, 6, 6, 5, 6, 7, 5, 6, 4, 8, 6, 8, 10};
    int responseFrequency[40];

    for (int z = 0; z < 40 ; z++){
        responseFrequency[z] = 0;
    }


    for (int i = 0; i < 40;i++){
        int responseNumber = responses[i];
        responseFrequency[responseNumber-1]++;
    }

    std::cout << "Response Summary:" << std::endl;

    for (int p = 1; p < 11; p++){
        std::cout << "Response " << p  << ":\t" << responseFrequency[p-1] << " Students"<<std::endl;
    }
    return 0;
}
```

The code first declares an int array named “responses” with 40 values. Next it initializes an int array named “responseFrequency”. After that it uses a “for” loop to set the “responseFrequency” values to zero. Next, it uses another “for” loop in order to check the number then it increments the value on that number’s index, for example if it detects 1 then it increments the value on responseFrequency[1-1]. After that it prints “Response Summary” with a line break. Next, it uses a for loop that prints “Response [value of p] [how many responded with that value] students” with a line break at the end. Lastly it returns 0

Output:

```
Response Summary:
Response 1:     2 Students
Response 2:     2 Students
Response 3:     2 Students
Response 4:     2 Students
Response 5:     5 Students
Response 6:     11 Students
Response 7:     5 Students
Response 8:     7 Students
Response 9:     1 Students
Response 10:    3 Students

--------------------------------
Process exited after 0.01563 seconds with return value 0
Press any key to continue . . .
```

8. Conclusion

In this lesson, I learned more about how to use for loops on arrays. The output part showed me with good examples on possible uses of arrays. Explaining the code made me look at these codes more in depth and not just skip them by copy-pasting them in a compiler. The supplementary activities challenged me more on creating logics for these programs. First, I need to make a code that checks the value then creates an asterisk based on how much the value of the numbers in the array. The next needs a code to check and keep track of how many responses have the same value.

9. Assessment Rubric