Assignment No. 4.3	
Pointers	
Course Code: CPE 007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 9/23/2025**
Section: CPE11S1	**Date Submitted: 9/23/2025**
Name(s): Cenar, Marqui Joshua	**Instructor: Engr. Jimlord M. Quejado**

6. Output

1. What is a pointer in C++?
- A pointer is a variable that stores the memory address of another variable instead of storing a different or actual value.
2. How does a pointer differ from a regular variable?
- Regular variables are the ones who keep its actual value like numbers, while a pointer keeps the location in memory where its value is stored.
3. What operator is used to get the address of a variable?
- An ampersand (&) symbol.
4. What operator is used to access the value stored at a pointer's address?
- The asterisk (*) symbol.
5. Why are pointers important in C++? Give two uses.
- Pointers are important because they are able to make programs more flexible. An example of its uses is Accessing memory addresses, this allows us to work with memory addresses which makes the programs more flexible. Another example is its cooperation or connection with arrays, since they act and have similarities with pointers.

7. Supplementary Activity

Identify the Output

1.

```
int x = 42;
int *ptr = &x;
cout << *ptr;
```

- This program will print out the value "42".

2.

```
int a = 5, b = 10;
int *p = &a;
p = &b;
cout << *p;
```

- The code will output the value "10".

3.

```
int arr[3] = {10, 20, 30};
int *p = arr;
cout << *p;
```

- This code will print out the first value of arr[3], which is “10”.

4.

```
int arr[4] = {2, 4, 6, 8};
int *p = arr;
p++;
cout << *p;
```

- The code will print out the second value of arr[4], which is “4” because the program contains “p++”

5.

```
int arr[3] = {5, 15, 25};
int *p = arr;
cout << *(p + 2);
```

- This program outputs the third value of arr[3], which is “25” because it contains a pointer (*p + 2).

Error Spotting

1.

```
int arr[3] = {1, 2, 3};
int *p = &arr;
```

- There should not be an ampersand (&) in “&arr”

2.

```
int arr[5];
int *p;
p = arr[2];
```

- “arr[2]” is an integer and you cannot assign an integer to a pointer. It should be “p = &arr[2]”.

3.

```
int arr[4] = {10, 20, 30, 40};
cout << *arr[2];
```

- “cout << *arr[2]” should not contain an asterisk or operator, It should be “cout << arr[2]”

8. Conclusion

From this assignment, I learned more about pointers and how they work and be used to access memory addresses in codes, also the values of each variable. I also learned that arrays and pointers are connected. I tried my best to understand what output will come out of each program and predict the right outcome. Then, the error spotting portion showed me how to identify wrong pointer declarations and how to avoid those mistakes. For this assignment, I think finished it fairly well, I didn't understand everything at first but looking at it again now, I see how each code has an error or I can somehow predict simple codes and its output like in this assignment.