

| Activity No. 4.3 | |
|---|---|
| Pointers | |
| Course Code: CPE 007 | Program: Computer Engineering |
| Course Title: Programming Logic and Design | Date Performed: 09/21/2025 |
| Section: CPE11S1 | Date Submitted: 09/23/2025 |
| Name(s): Canoy Hail B. | Instructor: Engr. Jimlord M. Quejado |
| 6. Output | |
| 7. Supplementary Activity | |
| <p>1.</p> <pre>int x = 42; int *ptr = &x; cout << *ptr;</pre> <p>Explanation :</p> <ul style="list-style-type: none"> - The output of this code is simple since the x is equal to 42 and the coded output is int *ptr = &x; that means print the value of x then cout << *ptr; is basically just to print x so the output will only be 42. <p>2.</p> <pre>int a = 5, b = 10; int *p = &a; p = &b; cout << *p;</pre> <p>Explanation :</p> <ul style="list-style-type: none"> - The output should print 10 because *p is equal to b although at first the pointer assigned int *p = &a; its cancelled because the next code the p is re assigned to &b then the program will print the value stored in p which is b and the value in b which is 10. <p>3.</p> <pre>int arr[3] = {10, 20, 30}; int *p = arr; cout << *p;</pre> <p>Explanation :</p> <ul style="list-style-type: none"> - The program should only output 10 because *p = arr and the program is asked to print *p it will only print the first value because there is no looping or increment value added in it. <p>4.</p> <pre>int arr[4] = {2, 4, 6, 8}; int *p = arr; p++; cout << *p;</pre> | |

Explanation :

- The program is said to print *p and the value of that is arr but since p++ adds an additional in the increment value so the value of p which is 0 = 2 will be move to 1 = 4 so the program will print 4.

5.

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

Explanation :

- So *p is equal to arr and the program wants to print *(p+2) which means the increment will go from 0 to 2 there for the program will print 25.

Error Spotting

Identify and fix the error(if any) in the codes below.

1.

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

Code :

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main () {  
5  
6      int arr[3] = {1, 2, 3};  
7      int *p = arr;  
8  
9  
10  
11      return 0;  
12  }
```

```

1  #include <iostream>
2  using namespace std;
3
4  int main () {
5
6  int arr[3] = {1, 2, 3};
7  int *p = &arr[0];
8
9
10
11     return 0;
12 }

```

Output :

```

-----
Process exited after 1.026 seconds with return value 0
Press any key to continue . . . |

```

Error & Explanation :

- The error is in the 6th line you can either use `int *p = arr;` or `int *p = &arr[0];` to point the value inside arr. although there is still no output because there is no cout for the value to be printed.

2.

```

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

```

Code :

```

1  #include <iostream>
2  using namespace std;
3
4  int main () {
5
6  int arr[5] = {11, 22, 33, 44, 55};
7  int *p;
8  p = &arr[2];
9
10
11      return 0;
12  }

```

Output :

```

-----
Process exited after 1.025 seconds with return value 0
Press any key to continue . . .

```

Error & Explanation :

- The code just needed to have the five value in arr and needed & to point the value in arr there is still no output because there is no cout for the value to be printed.

3.

```

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

```

Code :

```

1  #include <iostream>
2  using namespace std;
3
4  int main () {
5
6  int arr[4] = {10, 20, 30, 40};
7  cout << arr[2];
8
9
10
11      return 0;
12  }

```

Output :

```
30
-----
Process exited after 1.032 seconds with return value 0
Press any key to continue . . . |
```

Error & Explanation :

- The code just had * in arr which really isn't needed to print the 2nd value of arr.

8. Conclusion

- This activity, I learned more about pointers and operators, such as &, *, *p, p, and how to use them. This activity is easier compared to other activities, that's why I easily learned and adapted to it. I easily analyzed the code though it still took time, it's still good compared to other things I tried. I also was able to visualize code and even fix it sometimes just by seeing it. It's really important for pointing and moving values. This activity really helped me expand my visualization and analyzing skills, the way I look at it if the code can run just by seeing it. I did pretty well in this activity. It still might not be perfect but at least I know I'm growing and learning as a computer engineering student.