

Activity No. 5.2	
Hands-on Activity 5.2: Structures	
Course Code: CPE 007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: 10-02-25
Section: CPE11S1	Date Submitted: 10 - 04 -25
Name(s): Paula Esguerra	Instructor: Engr. Jimlord M. Quejado
6. Output	
Analysis of Sample Code of Using the structure member and structure pointer operators:	
<p>Struct variables are declared, the computer allocates memory for all the components. struct variables are accessed by using a selector. A selector is formed by placing a period (.). For example, the following code uses a.face for Ace and a.suit for Spades. Pointers can hold the address of another cell in memory, declared as aPtr then addresses a. When printing pointers, you could do three things: First is the variable (with the dot) itself, secondly, you could mix it up with dereference, then lastly you could use an arrow operator.</p>	
Analysis of Sample Code of Accessing Structure Members:	
<p>Struct variables are used to Books, then use string to title, author, subject and integer to book_id. Declare two Book variables: Books Book1 (Book1 as the title), Books Book2 (Book2 as the title). Then, a selector is used for identifying, for Book1 and Book2 therefore it was used to print the information that was programmed.</p>	
Sample Code of Structure Function Arguments	
<p>Struct variable is used to define Books, as also to string, string were used for title, author and subject, and integer for book_id. Declare Books 1 and 2, then Identify and put its site by using a selector. Then print its details.</p>	

7. Supplementary Activity

1. Create a program that uses a structure to store a rectangle's length and width. Write a function that accepts the structure as an argument and computes the area and perimeter of the rectangle.

```
#include <iostream>

struct RL {
    double length;
    double width;
};

void CRLP(const RL& rect) {
    double area = rect.length * rect.width;
    double perimeter = 2 * (rect.length + rect.width);

    std::cout << "-----+" << std::endl;
    std::cout << "Area: " << area << std::endl;
    std::cout << "Perimeter: " << perimeter << std::endl;
}

int main () {
    RL rect;

    std::cout << "Enter the length of the rectangle: ";
    std::cin >> rect.length;
    std::cout << "Enter the width of the rectangle: ";
    std::cin >> rect.width;

    if (rect.length <= 0 || rect.width <= 0 ) {
        std::cout << std::endl;
        std::cout << "-Length and width must be positive values!-" << std::endl;
        return 1;
    }

    CRLP(rect);
    return 0;
}
```

```
Enter the length of the rectangle: 20
Enter the width of the rectangle: 85
-----
Area: 1700
Perimeter: 210

-----
Process exited after 3.728 seconds with return value 0
Press any key to continue . . . |
```

```
Enter the length of the rectangle: -85
Enter the width of the rectangle: -78

-Length and width must be positive values!-

-----
Process exited after 3.636 seconds with return value 1
Press any key to continue . . . |
```

Analysis:

Struct rectangle, then **double** length and weight so it can identify decimals. Then use **void** so it doesn't return any value, create a parameter list, this function takes one argument. **const** ensures the functions can avoid copying the entire structure. Create an **If/else statement** to check if either the length or width is less than or equal to zero, or else it will return to code 1 (error). And lastly, create your print.

2. Write a program that creates a function multiple that determines if the integer entered from a keyboard is a multiple of some integer x.

```
1 #include <iostream>
2
3 bool multiple(int no1, int no2) {
4     if (no2 == 0) {
5         return false;
6     }
7     return no1 % no2 == 0;
8 }
9
10 int main() {
11     int no1, no2;
12
13     std::cout << "Enter the first integer: ";
14     std::cin >> no1;
15     std::cout << "Enter the second integer: ";
16     std::cin >> no2;
17
18     if (multiple(no1, no2)) {
19         std::cout << "-----*-----" << std::endl;
20         std::cout << no1 << " is a multiple of " << no2 << "." << std::endl;
21     } else {
22         std::cout << no1 << " is not a multiple of " << no2 << "." << std::endl;
23     }
24
25     return 0;
26 }
```

```
Enter the first integer: 90
Enter the second integer: 00
90 is not a multiple of 0.
```

```
-----  
Process exited after 3.964 seconds with return value 0  
Press any key to continue . . .
```

```
Enter the first integer: 20
Enter the second integer: 2
*-----*
20 is a multiple of 2.
```

```
-----  
Process exited after 5.637 seconds with return value 0
Press any key to continue . . .
```

Analysis:

Bool multiple defines a function that takes two integer arguments and returns a bool value indicating whether no1 or no2. If no2 is 0 then it returns 0 as false to prevent errors. Then **if/else statement** calls the functions with no1 and no2 as arguments, if it is true, then it executes the message for true, then if it is false, then it prints for false.

8. Conclusion

Here I observed that the activity done in the output section is very similar to each other, where they use the pointer, the selector, the struct and etc. Where instead they use the name itself, they put in a address, in which now hold the title, then the pointer is now we are using for printing the code. Then I observed in Supplementary section is we can use Bool variable, then the if/else statement is very much used to check every code before it executes. I played a bit in printing the output, where they have “+-----” as cut line paragraph which is very helpful to see which and which very cleanly.