

Activity No. 5.2	
Structures	
<b>Course Code:</b> CPE 007	<b>Program:</b> Computer Engineering
<b>Course Title:</b> Programming Logic and Design	<b>Date Performed:</b> 09/30/2025
<b>Section:</b> CPE11S1	<b>Date Submitted:</b> 10/04/2025
<b>Name(s):</b> Canoy Hail B.	<b>Instructor:</b> Engr. Jimlord M. Quejado
<b>6. Output</b>	
<p><b>Code :</b></p> <pre>#include &lt;iostream&gt; #include &lt;string&gt; using namespace std;  struct Card {     string face;     string suit; };  int main() {     Card a;      // structure variable     Card* aPtr; // structure pointer      // Assign values     a.face = "Ace";     a.suit = "Spades";      // Pointer points to structure 'a'     aPtr = &amp;a;      // Accessing members:     // Using the dot operator (.)     cout &lt;&lt; a.face &lt;&lt; " of " &lt;&lt; a.suit &lt;&lt; endl;      // Using the arrow operator (-&gt;)     cout &lt;&lt; aPtr-&gt;face &lt;&lt; " of " &lt;&lt; aPtr-&gt;suit &lt;&lt; endl;      // Using dereference (*) and dot     cout &lt;&lt; (*aPtr).face &lt;&lt; " of " &lt;&lt; (*aPtr).suit &lt;&lt; endl; }  return 0;</pre> <p><b>Sample Output:</b></p> <div style="background-color: black; color: white; padding: 10px; text-align: center;">         Ace of Spades          Ace of Spades          Ace of Spades     </div>	
<b>Explanation :</b>	

- So basically struct for creating a storage or blue print for the ace and spades then string face and string suit to create varieties of card for later. .then card a and card\*ptr for creating a structure and pointer then a.face and a.suit for adding value to face and suit. then aPtr = &a to make aPtr value the same as a. Then the easy part is to print it all out, `cout << a.face << " of " << a.suit << endl;` to print Ace of Spades because a.face is Ace and cout of then a.suit is Spades same as `cout << aPtr->face << " of " << aPtr->suit << endl;` since a.face is pointed at a.face it will print Ace then the other one is pointing to sui which means it will print suit. Then the last one `cout << (*aPtr).face << " of " << (*aPtr).suit << endl;` the Ace is `(*aPtr).face` then the Spades is `(*aPtr).suit`, then it should print Ace of Spades three times.

### Code :

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

// Define the structure
struct Books {
    string title;
    string author;
    string subject;
    int book_id;
};

int main() {
    // Declare two Book variables
    Books Book1;
    Books Book2;

    // Book 1 specification
    Book1.title = "C Programming";
    Book1.author = "Nuha Ali";
    Book1.subject = "C Programming Tutorial";
    Book1.book_id = 6495407;

    // Book 2 specification
    Book2.title = "Telecom Billing";
    Book2.author = "Zara Ali";
    Book2.subject = "Telecom Billing Tutorial";
    Book2.book_id = 6495700;

    // Print Book 1 info
    cout << "Book 1 title : " << Book1.title << endl;
    cout << "Book 1 author : " << Book1.author << endl;
    cout << "Book 1 subject : " << Book1.subject << endl;
    cout << "Book 1 book_id : " << Book1.book_id << endl;

    cout << endl; // for spacing

    // Print Book 2 info
    cout << "Book 2 title : " << Book2.title << endl;
```

```

cout << "Book 2 author : " << Book2.author << endl;
cout << "Book 2 subject : " << Book2.subject << endl;
cout << "Book 2 book_id : " << Book2.book_id << endl;
return 0;
}

```

Output:

```

Book 1 title    : C++ Programming
Book 1 author   : Nuha Ali
Book 1 subject  : C Programming Tutorial
Book 1 book_id  : 6495407

Book 2 title    : Telecom Billing
Book 2 author   : Zara Ali
Book 2 subject  : Telecom Billing Tutorial
Book 2 book_id  : 6495700

```

### Explanation :

- `#include <iostream> #include <string> using namespace std;` to include output and input functions, allow string and to not write `std::` in every `cout`. `struct Books` to contain the structure or values of books while `string title; string author; string subject; int book_id;` } is the value inside the book. `Books Book1; Books Book2;` to have 2 books, `Book1.title = "C Programming"; Book1.author = "Nuha Ali"; Book1.subject = "C Programming Tutorial"; Book1.book_id = 6495407;` to have the value and store it for book1 then for book 2 we do the same. `cout << "Book 1 title : " << Book1.title << endl; cout << "Book 1 author : " << Book1.author << endl; cout << "Book 1 subject : " << Book1.subject << endl; cout << "Book 1 book_id : " << Book1.book_id << endl; cout << endl;` to print the title, author, subject and the book id of book1 then make another one again for book2, then end with `return 0;`.
- 

### Code :

```

#include <iostream>
#include <string>
using namespace std;

// Define a structure
struct Books {
    string title;
    string author;
    string subject;
    int book_id;
};

// Function declaration (structure passed by value)
void printBook(Books book);

```

```

int main() {
    // Declare two Book variables
    Books Book1;
    Books Book2;

    // Book 1 specification
    Book1.title = "C Programming";
    Book1.author = "Nuha Ali";
    Book1.subject = "C Programming Tutorial";
    Book1.book_id = 6495407;

    // Book 2 specification
    Book2.title = "Telecom Billing";
    Book2.author = "Zara Ali";
    Book2.subject = "Telecom Billing Tutorial";
    Book2.book_id = 6495700;

    // Print details by passing the structure to a function
    printBook(Book1);
    cout << endl; // just for spacing
    printBook(Book2);

    return 0;
}

// Function definition
void printBook(Books book) {
    cout << "Book title : " << book.title << endl;
    cout << "Book author : " << book.author << endl;
    cout << "Book subject : " << book.subject << endl;
    cout << "Book book_id : " << book.book_id << endl;
}

```

Output:

```

Book title : C Programming
Book author : Nuha Ali
Book subject : C Programming Tutorial
Book book_id : 6495407

Book title : Telecom Billing
Book author : Zara Ali
Book subject : Telecom Billing Tutorial
Book book_id : 6495700

```

**Explanation :** So basically this programs starts with struct books and strings the title author and subject to to group the related data into one for later then we are gonna do void print book because void wont return any value and it will declare the function print book, then the main function, followed by book 1 and book2 as the variety of books then we are gonna put the data for the book1 and repeat for book2 and then we cout it, cout it then return 0;

## 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.

Code :

```
1 #include <iostream>
2 using namespace std;
3
4 void greet() {
5     cout << "-- Welcome to Rectangle Calculator --" << endl;
6 }
7
8
9 int area(int length, int width) {
10     int result = length * width;
11     return result;
12 }
13
14
15 int perimeter(int length, int width) {
16     int result = 2 * (length + width);
17     return result;
18 }
19
20 int main() {
21     greet();
22
23     int length, width;
24     cout << "Enter the length of the rectangle: ";
25     cin >> length;
26     cout << "Enter the width of the rectangle: ";
27     cin >> width;
28
29     cout << endl;
30
31     int areaResult = area(length, width);
32     cout << "Area of Rectangle: " << areaResult << endl;
33
34     int perimeterResult = perimeter(length, width);
35     cout << "Perimeter of Rectangle: " << perimeterResult << endl;
36
37     cout << endl;
38
39     return 0;
40 }
```

**Output :**

```
-- Welcome to Rectangle Calculator --
Enter the length of the rectangle: 90
Enter the width of the rectangle: 48

Area of Rectangle: 4320
Perimeter of Rectangle: 276
```

---

```
Process exited after 5.879 seconds with return value 0
Press any key to continue . . . |
```

**Explanation :**

- So this programs works first with struct to hold two pieces of data which is length and width and we encode how to compute the area first which is int area(int length, int width) { int result = length \* width; return result; then do one for the perimeter which is int perimeter(int length, int width) { int result = 2 \* (length + width); return result. then we use void because it doesn't return anything then greet for it to print the greet cout << Welcome to Rectangle Calculator. then we are gonna cout everything after that. then w int areaResult = area(length, width); and cout for area result then do it again for perimeter.

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.**

### Code :

```
1 #include <iostream>
2 using namespace std;
3
4 void multiple(int num, int x) {
5     if (num % x == 0) {
6         cout << num << " is a multiple of " << x << "." << endl;
7     } else {
8         cout << num << " is NOT a multiple of " << x << "." << endl;
9     }
10 }
11
12 int main() {
13     int num, x;
14
15     cout << "Enter a number: ";
16     cin >> num;
17
18     cout << "Enter another number to check if it's a multiple of: ";
19     cin >> x;
20
21     multiple(num, x);
22
23     return 0;
24 }
```

### Output :

```
Enter a number: 25
Enter another number to check if it's a multiple of: 5
25 is a multiple of 5.
```

```
-----
Process exited after 7.254 seconds with return value 0
Press any key to continue . . . |
```

### Explanation :

- so we start with void with function name multiple then two integers which is x and num then if (num % x == 0) to check if num divided by x has no remainder, then we use cout to print it as a multiple of x then else if it has a remainder it will print its is not a remainder of the multiple of the user entered. then main function then int num and x then cout for the user to have and read the instruction to enter a number and another to check if its a multiple of that number. then return 0;.

### 8. Conclusion

- In this activity I learned a lot, how to use functions better, how to organize them better. This activity is easier compared to others but it still took me a lot of time because I keep making errors. I learned a lot like using void, greet and struct, and overall I think I did good in this activity although it still hard for me to understand It I'm starting to get confident in my skills although it still is not enough because I got a bad memorization skills and still for me its still hard to code from scratch because if I forgot one code it will mean the all my efforts from the code

will mean nothing, I need to focus more on practice and memorizing the code so I can code better without reference.