CCC 204 Data Structures and Algorithms LABORATORY REPORT :
LAB 6# - Structs

Samano, Rajel Johann
Information Technology Program, CABECS
Colegio San Agustin – Bacolod

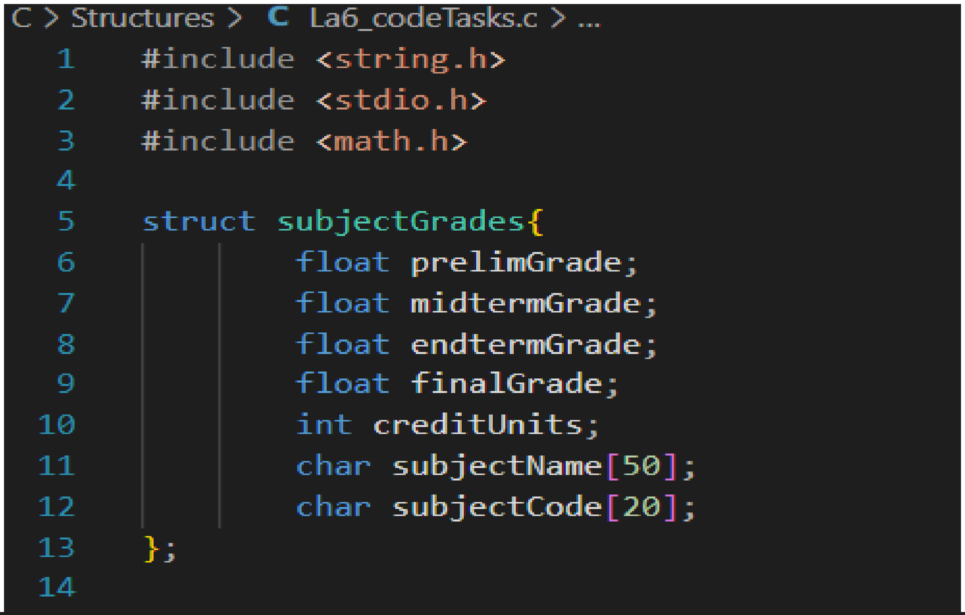
I. INTRODUCTION

For Laboratory Activity Number Six is to follow the objectives of the exercises and answer questions. The Three main objectives as follows:

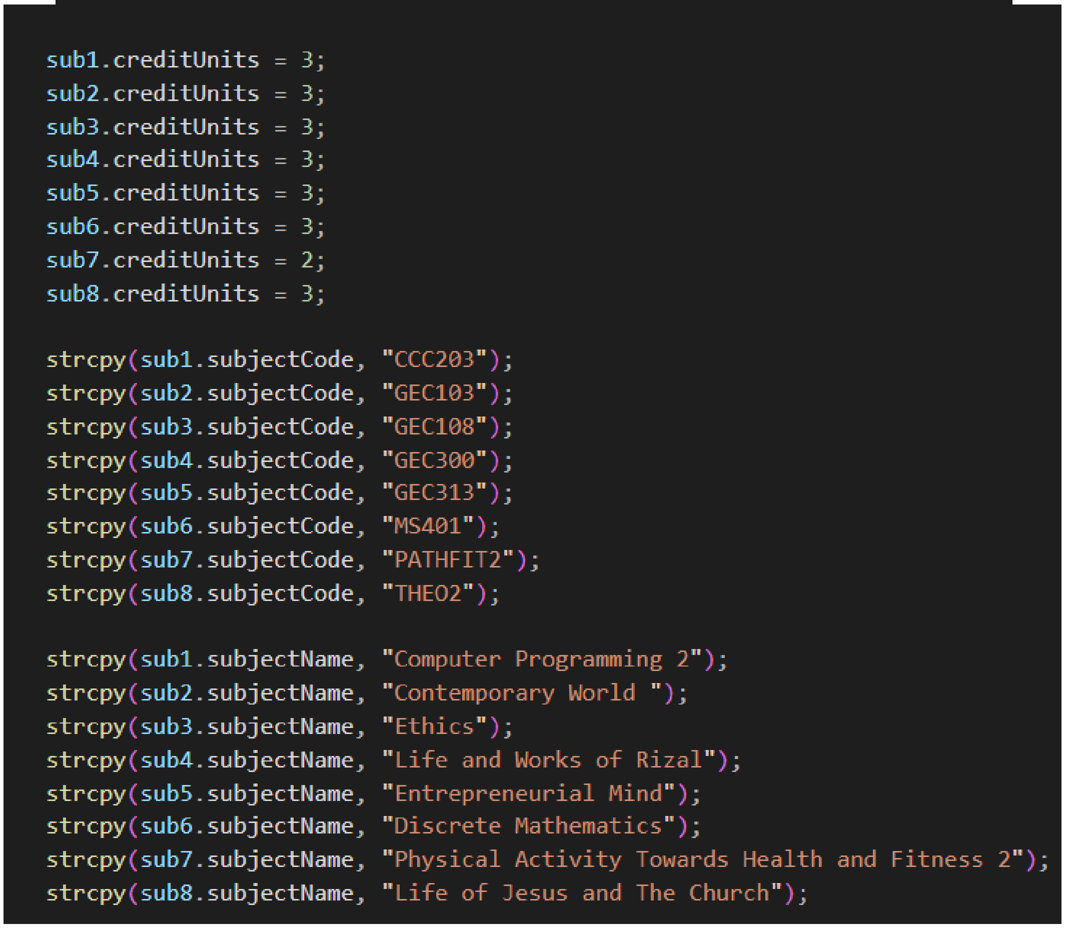
- **Describe struct and struct members**
- **Use struct and access struct members in code**
- **Perform operations and functions in struct**

II. IMPLEMENTATION / APPROACH

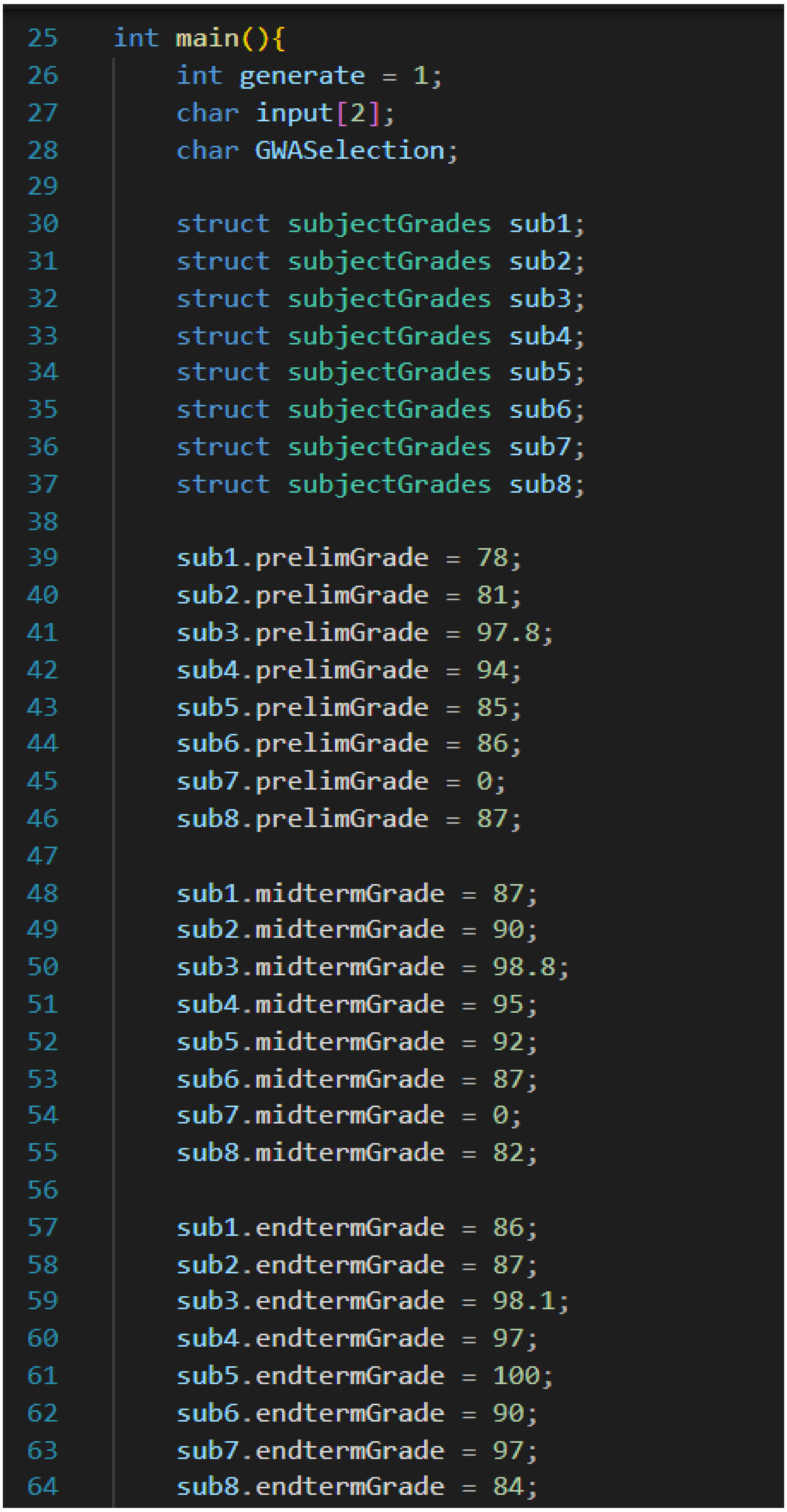
Figure 1-5. LA6_codeTasks.c with output



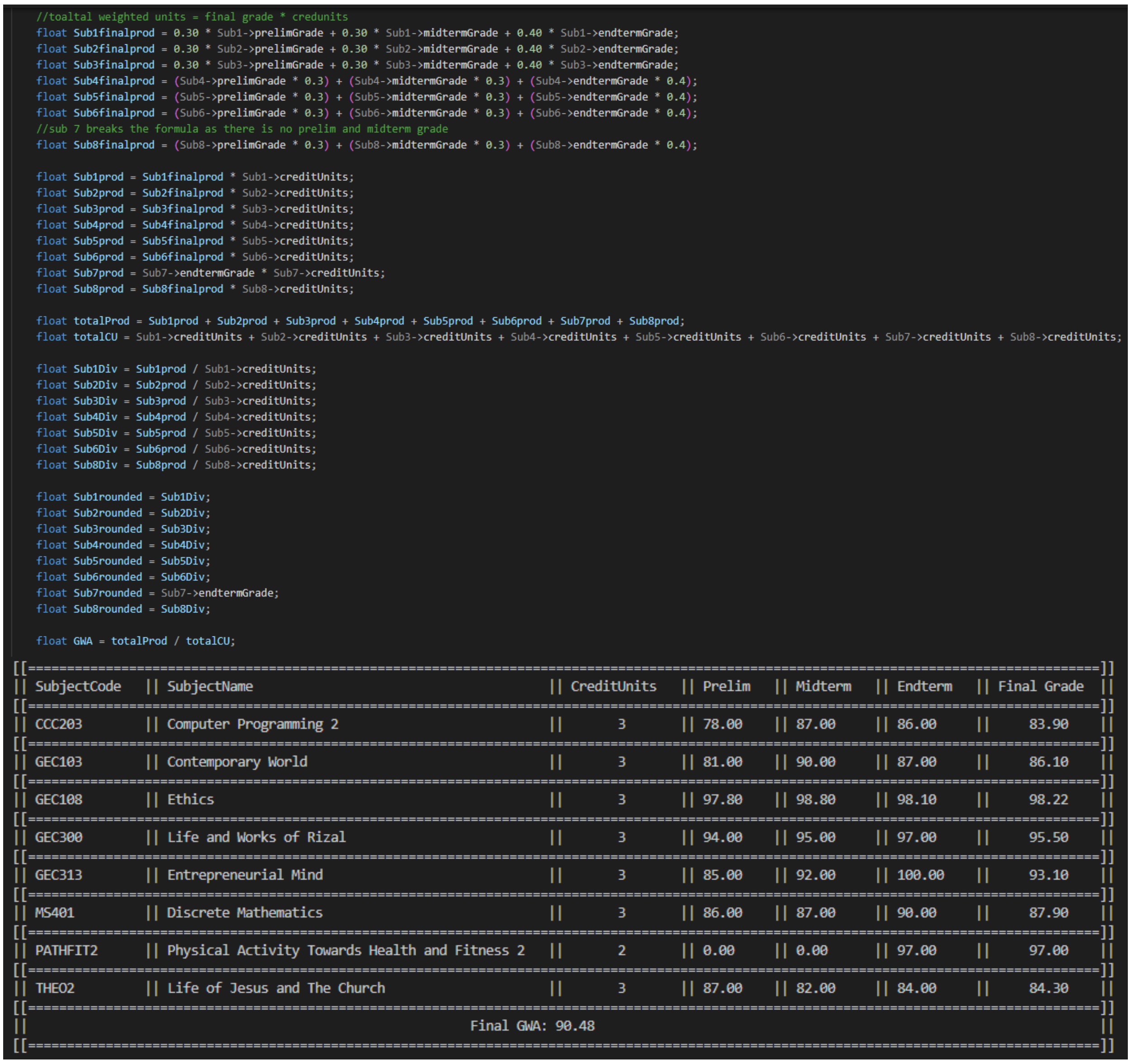
```c
#include <string.h>
#include <stdio.h>
#include <math.h>

struct subjectGrades{
        float prelimGrade;
        float midtermGrade;
        float endtermGrade;
        float finalGrade;
        int creditUnits;
        char subjectName[50];
        char subjectCode[20];
};

```



```c
sub1.creditUnits = 3;
sub2.creditUnits = 3;
sub3.creditUnits = 3;
sub4.creditUnits = 3;
sub5.creditUnits = 3;
sub6.creditUnits = 3;
sub7.creditUnits = 2;
sub8.creditUnits = 3;

strcpy(sub1.subjectCode, "CCC203");
strcpy(sub2.subjectCode, "GEC103");
strcpy(sub3.subjectCode, "GEC108");
strcpy(sub4.subjectCode, "GEC300");
strcpy(sub5.subjectCode, "GEC313");
strcpy(sub6.subjectCode, "MS401");
strcpy(sub7.subjectCode, "PATHFIT2");
strcpy(sub8.subjectCode, "THEO2");

strcpy(sub1.subjectName, "Computer Programming 2");
strcpy(sub2.subjectName, "Contemporary World ");
strcpy(sub3.subjectName, "Ethics");
strcpy(sub4.subjectName, "Life and Works of Rizal");
strcpy(sub5.subjectName, "Entrepreneurial Mind");
strcpy(sub6.subjectName, "Discrete Mathematics");
strcpy(sub7.subjectName, "Physical Activity Towards Health and Fitness 2");
strcpy(sub8.subjectName, "Life of Jesus and The Church");
```



```c
int main(){
    int generate = 1;
    char input[2];
    char GWASelection;

    struct subjectGrades sub1;
    struct subjectGrades sub2;
    struct subjectGrades sub3;
    struct subjectGrades sub4;
    struct subjectGrades sub5;
    struct subjectGrades sub6;
    struct subjectGrades sub7;
    struct subjectGrades sub8;

    sub1.prelimGrade = 78;
    sub2.prelimGrade = 81;
    sub3.prelimGrade = 97.8;
    sub4.prelimGrade = 94;
    sub5.prelimGrade = 85;
    sub6.prelimGrade = 86;
    sub7.prelimGrade = 0;
    sub8.prelimGrade = 87;

    sub1.midtermGrade = 87;
    sub2.midtermGrade = 90;
    sub3.midtermGrade = 98.8;
    sub4.midtermGrade = 95;
    sub5.midtermGrade = 92;
    sub6.midtermGrade = 87;
    sub7.midtermGrade = 0;
    sub8.midtermGrade = 82;

    sub1.endtermGrade = 86;
    sub2.endtermGrade = 87;
    sub3.endtermGrade = 98.1;
    sub4.endtermGrade = 97;
    sub5.endtermGrade = 100;
    sub6.endtermGrade = 90;
    sub7.endtermGrade = 97;
    sub8.endtermGrade = 84;
```



```
    //toaltal weighted units = final grade * credunits
    float Sub1finalprod = 0.30 * Sub1->prelimGrade + 0.30 * Sub1->midtermGrade + 0.40 * Sub1->endtermGrade;
    float Sub2finalprod = 0.30 * Sub2->prelimGrade + 0.30 * Sub2->midtermGrade + 0.40 * Sub2->endtermGrade;
    float Sub3finalprod = 0.30 * Sub3->prelimGrade + 0.30 * Sub3->midtermGrade + 0.40 * Sub3->endtermGrade;
    float Sub4finalprod = (Sub4->prelimGrade * 0.3) + (Sub4->midtermGrade * 0.3) + (Sub4->endtermGrade * 0.4);
    float Sub5finalprod = (Sub5->prelimGrade * 0.3) + (Sub5->midtermGrade * 0.3) + (Sub5->endtermGrade * 0.4);
    float Sub6finalprod = (Sub6->prelimGrade * 0.3) + (Sub6->midtermGrade * 0.3) + (Sub6->endtermGrade * 0.4);
    //sub 7 breaks the formula as there is no prelim and midterm grade
    float Sub8finalprod = (Sub8->prelimGrade * 0.3) + (Sub8->midtermGrade * 0.3) + (Sub8->endtermGrade * 0.4);

    float Sub1prod = Sub1finalprod * Sub1->creditUnits;
    float Sub2prod = Sub2finalprod * Sub2->creditUnits;
    float Sub3prod = Sub3finalprod * Sub3->creditUnits;
    float Sub4prod = Sub4finalprod * Sub4->creditUnits;
    float Sub5prod = Sub5finalprod * Sub5->creditUnits;
    float Sub6prod = Sub6finalprod * Sub6->creditUnits;
    float Sub7prod = Sub7->endtermGrade * Sub7->creditUnits;
    float Sub8prod = Sub8finalprod * Sub8->creditUnits;

    float totalProd = Sub1prod + Sub2prod + Sub3prod + Sub4prod + Sub5prod + Sub6prod + Sub7prod + Sub8prod;
    float totalCU = Sub1->creditUnits + Sub2->creditUnits + Sub3->creditUnits + Sub4->creditUnits + Sub5->creditUnits + Sub6->creditUnits + Sub7->creditUnits + Sub8->creditUnits;

    float Sub1Div = Sub1prod / Sub1->creditUnits;
    float Sub2Div = Sub2prod / Sub2->creditUnits;
    float Sub3Div = Sub3prod / Sub3->creditUnits;
    float Sub4Div = Sub4prod / Sub4->creditUnits;
    float Sub5Div = Sub5prod / Sub5->creditUnits;
    float Sub6Div = Sub6prod / Sub6->creditUnits;
    float Sub8Div = Sub8prod / Sub8->creditUnits;

    float Sub1rounded = Sub1Div;
    float Sub2rounded = Sub2Div;
    float Sub3rounded = Sub3Div;
    float Sub4rounded = Sub4Div;
    float Sub5rounded = Sub5Div;
    float Sub6rounded = Sub6Div;
    float Sub7rounded = Sub7->endtermGrade;
    float Sub8rounded = Sub8Div;

    float GWA = totalProd / totalCU;
```

```
[[=========================================================================================================]]
|| SubjectCode   || SubjectName                              || CreditUnits  || Prelim  || Midterm  || Endterm  || Final Grade  ||
[[=========================================================================================================]]
|| CCC203        || Computer Programming 2                   ||       3      || 78.00   || 87.00    || 86.00    ||     83.90    ||
[[=========================================================================================================]]
|| GEC103        || Contemporary World                       ||       3      || 81.00   || 90.00    || 87.00    ||     86.10    ||
[[=========================================================================================================]]
|| GEC108        || Ethics                                   ||       3      || 97.80   || 98.80    || 98.10    ||     98.22    ||
[[=========================================================================================================]]
|| GEC300        || Life and Works of Rizal                  ||       3      || 94.00   || 95.00    || 97.00    ||     95.50    ||
[[=========================================================================================================]]
|| GEC313        || Entrepreneurial Mind                     ||       3      || 85.00   || 92.00    || 100.00   ||     93.10    ||
[[=========================================================================================================]]
|| MS401         || Discrete Mathematics                     ||       3      || 86.00   || 87.00    || 90.00    ||     87.90    ||
[[=========================================================================================================]]
|| PATHFIT2      || Physical Activity Towards Health and Fitness 2 ||  2   || 0.00    || 0.00     || 97.00    ||     97.00    ||
[[=========================================================================================================]]
|| THEO2         || Life of Jesus and The Church             ||       3      || 87.00   || 82.00    || 84.00    ||     84.30    ||
[[=========================================================================================================]]
||                                                   Final GWA: 90.48                                            ||
[[=========================================================================================================]]
```

Implement struct to store subjects and compute for GWA.

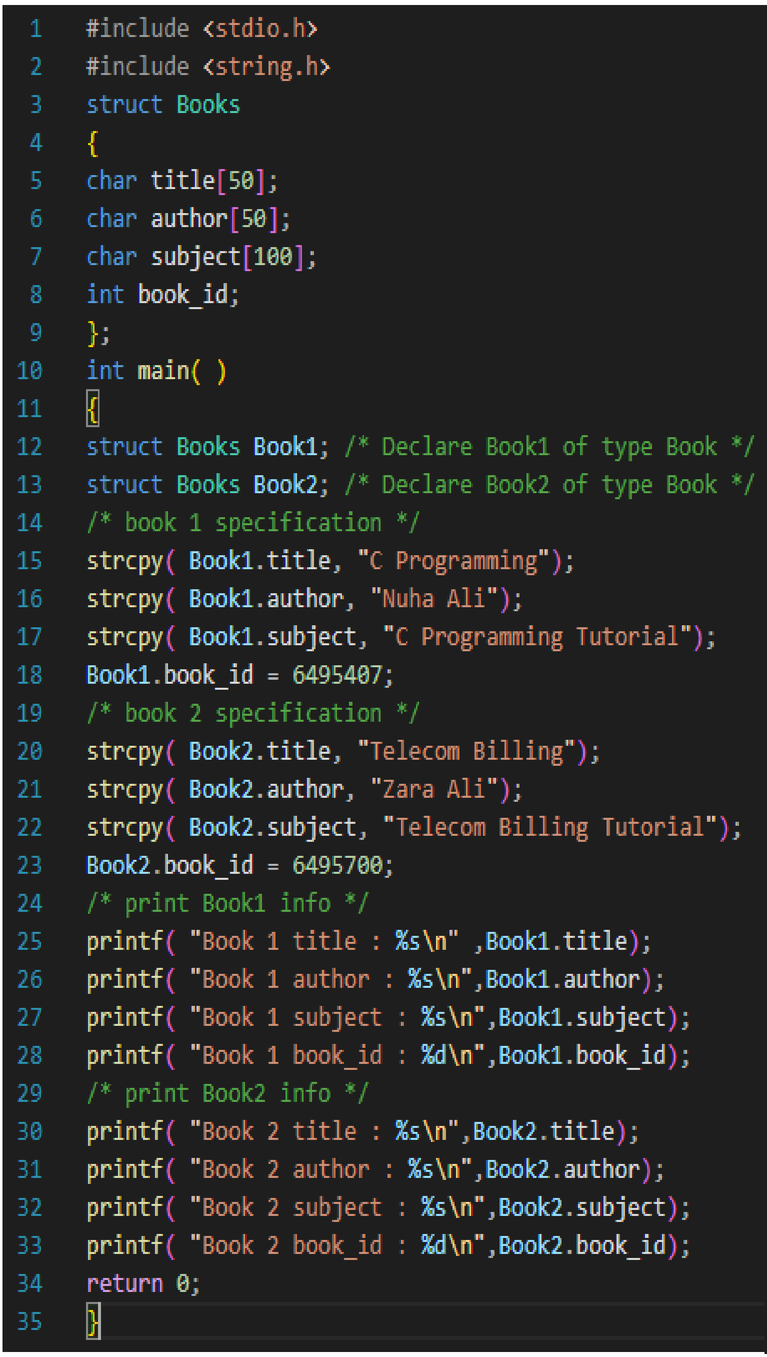
Struct name: subjectGrades Struct members:
1. subjectName
2. subjectCode
3. prelimGrade
4. midtermGrade
5. endtermGrade
6. finalGrade
7. creditUnits

Functions/methods to be used with struct: computeGWA (term)
Additional details: Final grade computation = 30% Prelim + 30% Midterm + 40% Endterm.

My approach towards this task was first to modify the code that we did on our class about struct and have subjectName, subjectCode, prelimGrade, midtermGrade, endtermGrade, finalGrade, and creditUnits and then after that was to check my aims grades for an easier reference to follow so I just did it simply after asking my sir on how to get the average or how to get GWA after that was done was to design it then just get some ideas in the example codes that were provided in the PDF file.

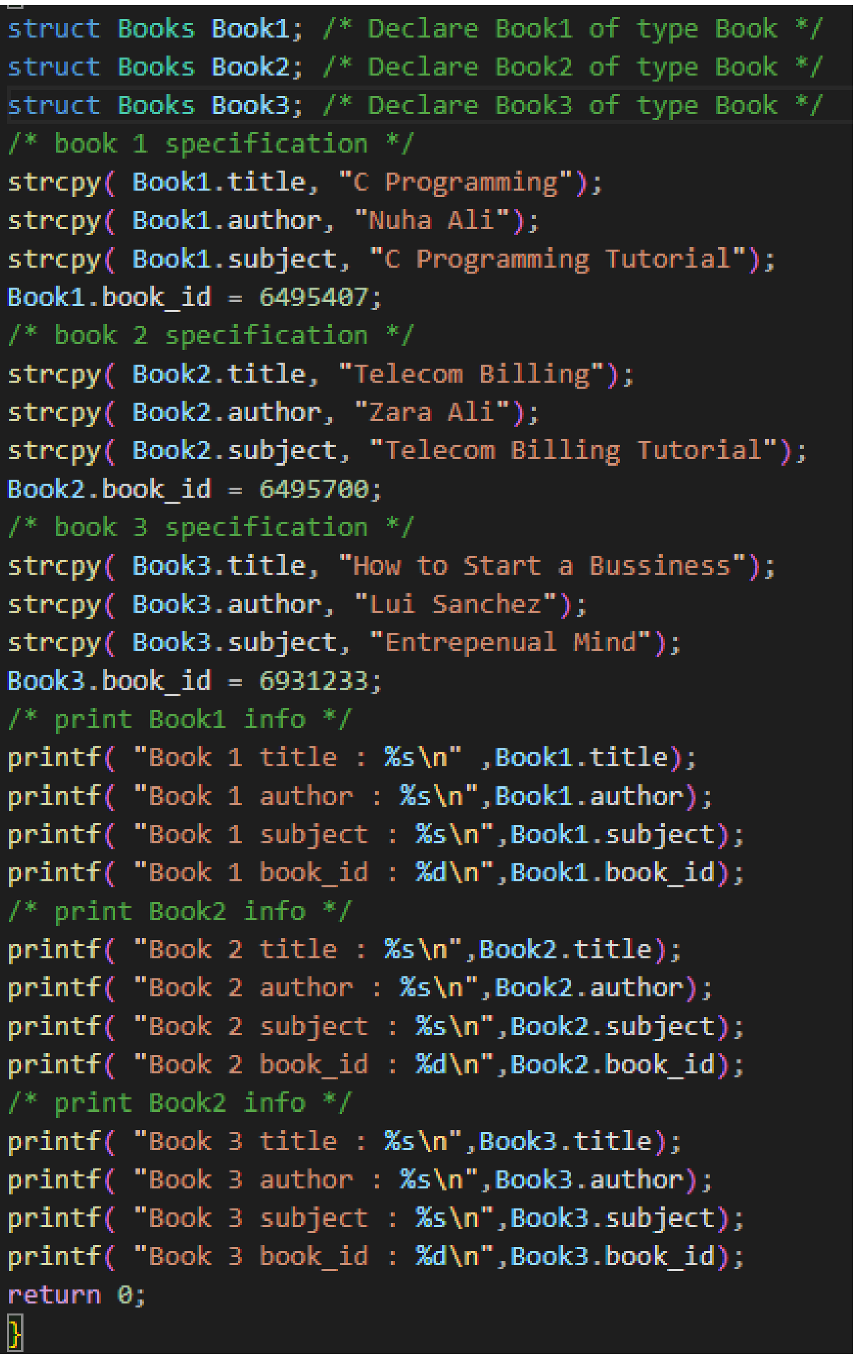
III. EXPERIMENTAL FINDINGS / DISCUSSIONS

Figure 6. Exercise1



```c
#include <stdio.h>
#include <string.h>
struct Books
{
char title[50];
char author[50];
char subject[100];
int book_id;
};
int main( )
{
struct Books Book1; /* Declare Book1 of type Book */
struct Books Book2; /* Declare Book2 of type Book */
/* book 1 specification */
strcpy( Book1.title, "C Programming");
strcpy( Book1.author, "Nuha Ali");
strcpy( Book1.subject, "C Programming Tutorial");
Book1.book_id = 6495407;
/* book 2 specification */
strcpy( Book2.title, "Telecom Billing");
strcpy( Book2.author, "Zara Ali");
strcpy( Book2.subject, "Telecom Billing Tutorial");
Book2.book_id = 6495700;
/* print Book1 info */
printf( "Book 1 title : %s\n" ,Book1.title);
printf( "Book 1 author : %s\n",Book1.author);
printf( "Book 1 subject : %s\n",Book1.subject);
printf( "Book 1 book_id : %d\n",Book1.book_id);
/* print Book2 info */
printf( "Book 2 title : %s\n",Book2.title);
printf( "Book 2 author : %s\n",Book2.author);
printf( "Book 2 subject : %s\n",Book2.subject);
printf( "Book 2 book_id : %d\n",Book2.book_id);
return 0;
}
```

Figure7. Exercise1 with Book3



```c
struct Books Book1; /* Declare Book1 of type Book */
struct Books Book2; /* Declare Book2 of type Book */
struct Books Book3; /* Declare Book3 of type Book */
/* book 1 specification */
strcpy( Book1.title, "C Programming");
strcpy( Book1.author, "Nuha Ali");
strcpy( Book1.subject, "C Programming Tutorial");
Book1.book_id = 6495407;
/* book 2 specification */
strcpy( Book2.title, "Telecom Billing");
strcpy( Book2.author, "Zara Ali");
strcpy( Book2.subject, "Telecom Billing Tutorial");
Book2.book_id = 6495700;
/* book 3 specification */
strcpy( Book3.title, "How to Start a Bussiness");
strcpy( Book3.author, "Lui Sanchez");
strcpy( Book3.subject, "Entrepenual Mind");
Book3.book_id = 6931233;
/* print Book1 info */
printf( "Book 1 title : %s\n" ,Book1.title);
printf( "Book 1 author : %s\n",Book1.author);
printf( "Book 1 subject : %s\n",Book1.subject);
printf( "Book 1 book_id : %d\n",Book1.book_id);
/* print Book2 info */
printf( "Book 2 title : %s\n",Book2.title);
printf( "Book 2 author : %s\n",Book2.author);
printf( "Book 2 subject : %s\n",Book2.subject);
printf( "Book 2 book_id : %d\n",Book2.book_id);
/* print Book2 info */
printf( "Book 3 title : %s\n",Book3.title);
printf( "Book 3 author : %s\n",Book3.author);
printf( "Book 3 subject : %s\n",Book3.subject);
printf( "Book 3 book_id : %d\n",Book3.book_id);
return 0;
}
```

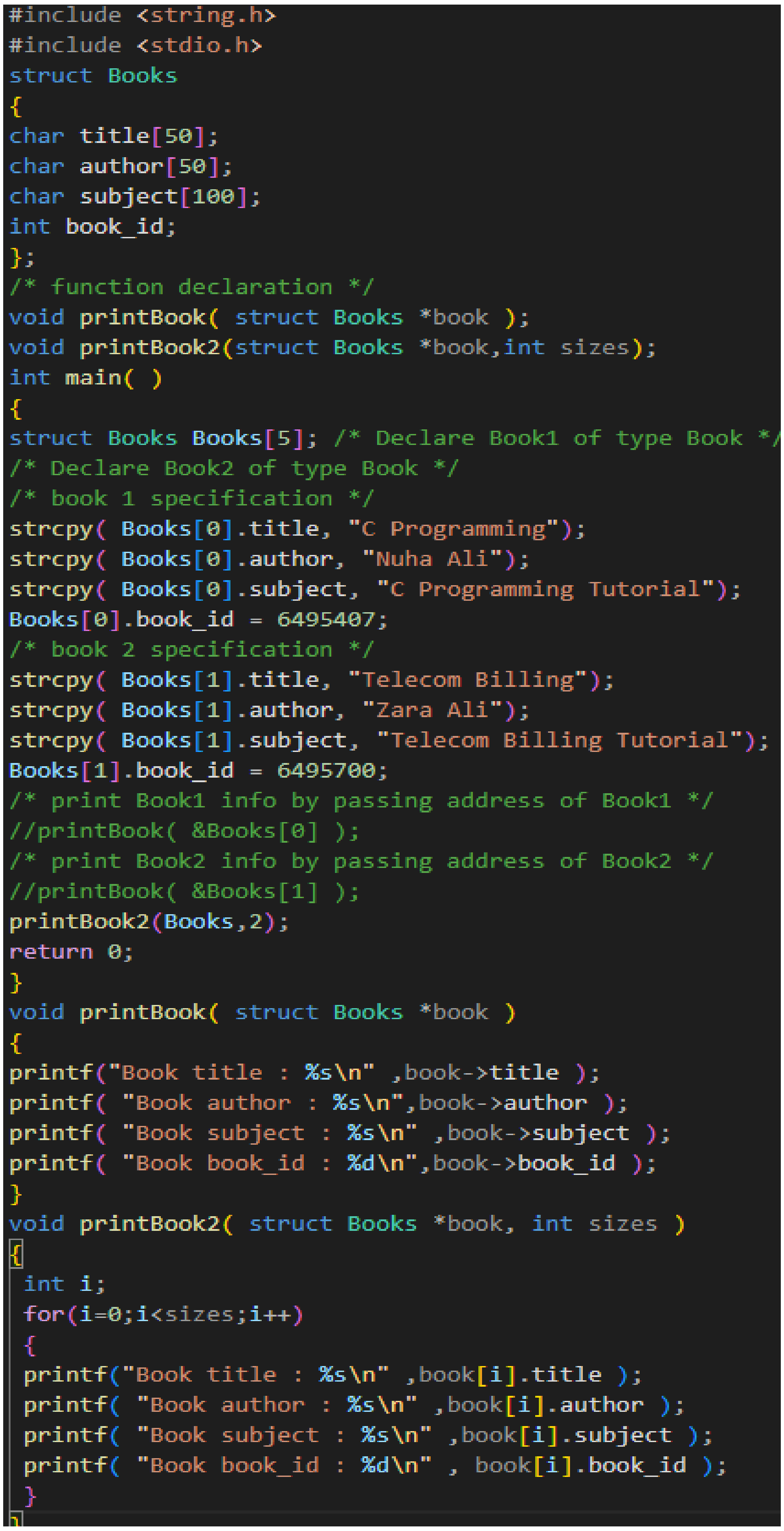
1. **What is the purpose of the line: struct Books Book1; ?**

Ans: The purpose for the line: struct Books Book1 is so to make Book1 an variable where it has the members from book like author, title, subject, and book_id.

2. **If an additional "Book" is to be created, what should be added to the code?**

Ans: To make another book or Book3 you need to create a new variable named Book3 and have the members from book3 be declared or filled like for the title and subject etc. an example would look like:

Figure 8. Excercise2



```c
#include <string.h>
#include <stdio.h>
struct Books
{
char title[50];
char author[50];
char subject[100];
int book_id;
};
/* function declaration */
void printBook( struct Books *book );
void printBook2(struct Books *book,int sizes);
int main( )
{
struct Books Books[5]; /* Declare Book1 of type Book */
/* Declare Book2 of type Book */
/* book 1 specification */
strcpy( Books[0].title, "C Programming");
strcpy( Books[0].author, "Nuha Ali");
strcpy( Books[0].subject, "C Programming Tutorial");
Books[0].book_id = 6495407;
/* book 2 specification */
strcpy( Books[1].title, "Telecom Billing");
strcpy( Books[1].author, "Zara Ali");
strcpy( Books[1].subject, "Telecom Billing Tutorial");
Books[1].book_id = 6495700;
/* print Book1 info by passing address of Book1 */
//printBook( &Books[0] );
/* print Book2 info by passing address of Book2 */
//printBook( &Books[1] );
printBook2(Books,2);
return 0;
}
void printBook( struct Books *book )
{
printf("Book title : %s\n" ,book->title );
printf( "Book author : %s\n",book->author );
printf( "Book subject : %s\n" ,book->subject );
printf( "Book book_id : %d\n",book->book_id );
}
void printBook2( struct Books *book, int sizes )
{
 int i;
 for(i=0;i<sizes;i++)
 {
 printf("Book title : %s\n" ,book[i].title );
 printf( "Book author : %s\n" ,book[i].author );
 printf( "Book subject : %s\n" ,book[i].subject );
 printf( "Book book_id : %d\n" , book[i].book_id );
 }
}
```

1. What situations is -> used over . for accessing struct members?

The dot (.) operator is used to access a member of a struct, while the arrow operator (->) in C is used to access a member of a struct which is referenced by the pointer in question.

IV. CONCLUSIONS

I have learned that you can group variable in what is called a construct when there are multiple variable being used it is much more easier and easy to understand which to me would still need some practice in or times where I am going to be using this more often would be on my pass time.

References:

https://stackoverflow.com/questions/13366083/why-does-the-arrow-operator-in-c-exist#

https://www.geeksforgeeks.org/structures-c/