FEU INSTITUTE OF TECHNOLOGY

COLLEGE OF COMPUTER STUDIES

CCS007L
(COMPUTER PROGRAMMING 2)

EXERCISE

4

STRUCTURES

Student Name / Group Name:	
Members (if Group):	**Name** / **Role** ( ) / ( ) ( ) / ( ) ( ) / ( )
Section:	
Professor:	

I. PROGRAM OUTCOME/S (PO) ADDRESSED BY THE LABORATORY EXERCISE

- Apply knowledge through the use of current techniques and tools necessary for the IT profession. [PO: I]

II. COURSE LEARNING OUTCOME/S (CLO)ADDRESSED BY THE LABORATORY EXERCISE

- Adapt and apply appropriate techniques, resources, and modern computing tools to create computing applications [CLO: 3]

III. INTENDED LEARNING OUTCOME/S (ILO) OF THE LABORATORY EXERCISE

At the end of this exercise, students must be able to:

- Create user- defined types using structures, access its elements and process
- Create a record management program.

IV. BACKGROUND INFORMATION

A **structure** is a group of elements which are of different type. Each of the elements is identified by its own identifier and they are called member.

The format for defining a structure is:

```
struct MyStructure {
  Members
};
```

A structure can be thought of as an object without any member functions. The important property of structures is that the data in a structure can be a collection of data items of diverse types.

A structure variable can hold values just like any other variable can. A structure value is a collection of smaller values called member values.

Here is an example program:

```
struct database {
  int id_number;
  int age;
  float salary;
};

int main()
{
  database employee;
  employee.age = 22;
  employee.id_number = 1;
  employee.salary = 12000.21;
}
```

Structures are used to represent a record, suppose you want to keep track of your books in a library. You might want to track the following attributes about each book:

- Title
- Author
- Subject
- Book ID

To access any member of a structure, we use the member access operator (the dot operator). The member access operator is coded as a period between the structure variable name and the structure member that we wish to access. You would use struct keyword to define variables of structure type.

V. GRADING SYSTEM/ RUBRIC

Trait	(Excellent)	(Good)	(Fair)	(Poor)
Requirement Specification(30pts)	Able to identify correctly all input and output and provide alternative. **(28-20pts)**	Able to identify correctly all input and output **(25-17pts)**	Able to identify only one input or output **(22-14pts)**	Unable to identify any input and output **(20-11pts)**
Data type(20pts)	Able to apply required data type or data structure and produce correct results **(18-20pts)**	Able to apply required data type or data structure and produce partially correct results **(15-17pts)**	Able to identify required data type or data structure but does apply correctly **(12-14pts)**	Unable to identify required data type **(9-11pts)**
Input Validation(20pts)	The program works and meets all specifications. Does exception al checking for errors and out-of- range data **(18-20pts)**	The program works and meets all specifications. Does some checking for errors and out of range data **(15-17pts)**	The program produces correct results but does not display correctly Does not check for errors and out of range data **(12-14pts)**	The program produce s incorrect results **(9-11pts)**
Free from syntax, logic, and runtime errors (10pts)	Unable to run program **(10pts)**	Able to run program but have logic error **(8-9pts)**	Able to run program correctly without any logic error and display inappropriate output **(6-7pts)**	Able to run program correctly without any logic error and display appropriate output **(5pts)**
Delivery (10pts)	The program was delivered on time **(10pts)**	The program was delivered after 5 minutes from the time required. **(8-9pts)**	The program was delivered after 10 minutes from the time required. **(6-7pts)**	The program was delivered after 15 (or more) minutes from the time required. **(5pts)**
Use of Comments (10pts)	Specific purpose is noted for each function, control structure, input requirements, and output results. **(10pts)**	Specific purpose is noted for each function and control structure. **(8-9pts)**	Purpose is noted for each function. **(6-7pts)**	No comments included. **(5pts)**

VI. LABORATORY ACTIVITY

INSTRUCTIONS:
Copy your source codes to be pasted in this document as well as a screen shot of your running output.

ACTIVITY 4.1: Player's Record

Write a C++ program to keep records and compute for the scores of 5 players. The information of each player contains: Nickname, Age and two best played scores.

The program will prompt the user to choose the operation of records from a menu as shown below:

```
==========================================
                MENU
==========================================
```

1. Add record
2. View players records
3. Compute for the average
4. Show the player(s) who gets the max average.
5. Show the player(s) who gets the min average.
6. Exit

VII. QUESTION AND ANSWER

Briefly answer the questions below. Avoid erasures. For group activity, specify the name of GROUP MEMBER/s who answered the question. Do not forget to include the source for all NON-ORIGINAL IDEAS.

• What are the ways to use or declare structure in your program?
• How do you access a member in a structure?

VIII. REFERENCES

- Zak, Dianne (2016). An Introduction to Programming with C++
- Deitel, Paul & Deitel, Harvey (2012). C++ How To Program, Eighth Edition
- https://www.cprogramming.com/tutorial/lesson7.html