FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CCS0007
Computer Programming 2 for IT

EXERCISE

3

Structures

CAPILI, MARK ANGELO	Joie Ann Maghanoy
1/22/20	1/22/20

I. OBJECTIVES

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

II. 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.

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.

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 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.

III. EXPERIMENTAL PROCEDURE

INSTRUCTIONS:

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

Upload your document using the link provided in your Canvas.

ACTIVITY 3.1: 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

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

int choice;
//  structure  for  storing  all  th
players data
struct Person
{
    char name[50];
    int age;
    int score1;
    int score2;
    float ave;
};

//for IU
void menu(){
   cout                                <<
"**********************************\n";
   cout  <<  "                        MENU
\n";
   cout                                <<
"**********************************\n";
   cout << "1. Add Record\n";
   cout    <<    "2.    View    Players
Records\n";
   cout   <<   "3.   Compute   For   The
Average\n";
   cout << "4. Show the player(s) who
get the max average\n";
   cout << "5. Show the player(s) who
get the min average\n";
   cout << "6. Exit\n";
   cout << "Enter the # of choice: ";
   cin >> choice;

}



// ALGORITHM FOR ALL THE REQUIREMENTS
in the MENU
```

```
**********************************
              MENU
**********************************
1. Add Record
2. View Players Records
3. Compute For The Average
4. Show the player(s) who get the max average
5. Show the player(s) who get the min average
6. Exit
Enter the # of choice:
```

```
**********************************
              MENU
**********************************
1. Add Record
2. View Players Records
3. Compute For The Average
4. Show the player(s) who get the max average
5. Show the player(s) who get the min average
6. Exit
Enter the # of choice: 1
PLAYER 1 RECORDS
Enter Name: MARK1
Enter Age: 18
Enter 1st score: 5
Enter 2nd score: 10
```

```
int main() {
   int temp1,temp2;
   int aveT = 0;
   menu();
   Person p[5];
   //  setting  all  the  members  to
default values
   for(int i = 0 ; i < 5 ; i++) {
      p[i].age = 0;
      p[i].score1 = 0;
      p[i].score2 = 0;
   }
   while(true){
      switch(choice) {
        //FOR STORING THE DATA INSIDE
tHE STRUCT
        case 1:
             for(int i = 0 ; i < 1 ;
i++) {
                cout << "PLAYER " << i
+ 1 << " RECORDS\n";
                cout << "Enter Name: ";
                cin >> p[i].name;
                cout << "Enter Age: ";
                cin >> p[i].age;
                cout   <<   "Enter   1st
score: ";
                cin >> p[i].score1;
                cout   <<   "Enter   2nd
score: ";
                cin >> p[i].score2;
                system ("cls");
             }
             menu();
          break;
          //    FOR    OUTPUTING    THE
AVERAGES
        case 2:
          //if  age  is  STILL  0  then
there's still no inputs
          if (p[0].age == 0) {
             cout  <<  "NO  INPUT"  <<
endl;
             system ("pause");
             menu();

             }
```

```
nter the # of choice: 2
LAYER 1 RECORDS:
ICKNAME: MARK1
GE: 18
CORE1: 5
CORE2: 10

LAYER 2 RECORDS:
ICKNAME: MARK2
GE: 18
CORE1: 15
CORE2: 20

LAYER 3 RECORDS:
ICKNAME: MARK3
GE: 18
CORE1: 30
CORE2: 35

LAYER 4 RECORDS:
ICKNAME: MARK4
GE: 18
CORE1: 45
CORE2: 50

LAYER 5 RECORDS:
ICKNAME: MARK5
GE: 18
CORE1: 50
CORE2: 60

ress any key to continue . . .
```

```
            for(int i = 0 ; i < 5 ;
i++) {

                cout << "PLAYER " <<
i+1 << " RECORDS: \n";
                cout << "NICKNAME: " <<
p[i].name << endl;
                cout << "AGE: " <<
p[i].age << endl;
                cout << "SCORE1: " <<
p[i].score1 << endl;
                cout << "SCORE2: " <<
p[i].score2 << endl << endl;
                p[i].ave = (p[i].score1
+ p[i].score2) / 2;
             }
             system ("pause");
             system ("cls");
             menu();
             break;
             //  TOTALLING  ALL  THE
AVERAGES
        case 3:
          //if age is STILL 0 then
there's still no inputs
          if (p[0].age == 0) {
             cout << "NO INPUT" <<
endl;
             system ("pause");
             menu();
          }
          for(int i = 0 ; i < 5 ; i++)
{
             cout << "PLAYER " <<
p[i].name << " SCORE AVERAGE: " <<
endl;
             cout << p[i].ave << endl
<< endl;
          }
          cout << "TOTAL 5 PLAYERS
AVERAGE: ";
          for(int i = 0 ; i < 5 ; i++)
{
             aveT = p[i].ave + aveT ;
          }
          cout << aveT / 5 << endl;
          system ("pause");
```

```
PLAYER MARK1 SCORE AVERAGE:
7

PLAYER MARK2 SCORE AVERAGE:
17

PLAYER MARK3 SCORE AVERAGE:
32

PLAYER MARK4 SCORE AVERAGE:
47

PLAYER MARK5 SCORE AVERAGE:
57

TOTAL 5 PLAYERS AVERAGE: 32
Press any key to continue . . .
```

```
**********************************
              MENU
**********************************
1. Add Record
2. View Players Records
3. Compute For The Average
4. Show the player(s) who get the max average
5. Show the player(s) who get the min average
6. Exit
Enter the # of choice: 5
THE LOWER AVERAGE AMONG ALL PLAYER IS : MARK1
Press any key to continue . . .
```

```
Enter the # of choice: 4
THE HIGHEST AVERAGE AMONG ALL PLAYER IS : MARK5
Press any key to continue . . .
```

```
            menu();
            break;
            //  ALGO  FOR  PRODUCING  THE
MAX AVERAGE
         case 4:
            //if  age  is  STILL  0  then
there's still no inputs
            if (p[0].age == 0) {
               cout  <<  "NO  INPUT"  <<
endl;
               system ("pause");
               menu();
            }
            temp1 = p[0].ave;
            for(int i = 0 ; i < 5 ; i++)
{
               if(temp1 < p[i].ave){
                  temp1 = p[i].ave;
               }
            }
            cout << "THE HIGHEST AVERAGE
AMONG ALL PLAYER IS : ";
            for(int i = 0 ; i < 5 ; i++)
{
               if(temp1 == p[i].ave){
                  cout  <<  p[i].name  <<
endl;
                  break;
               }
            }
            system ("pause");
            menu();
            break;
            //ALGO FOR PRODUCING THE MIN
AVERAGE
         case 5:
            //if  age  is  STILL  0  then
there's still no inputs
            if (p[0].age == 0) {
               cout  <<  "NO  INPUT"  <<
endl;
               system ("pause");
               menu();
            }
            temp2 = p[0].ave;
            for(int i = 0 ; i < 5 ; i++)
{
```

```
**********************************
              MENU
**********************************
1. Add Record
2. View Players Records
3. Compute For The Average
4. Show the player(s) who get the max average
5. Show the player(s) who get the min average
6. Exit
Enter the # of choice: 6

--------------------------------
Process exited after 172.8 seconds with return value 0
Press any key to continue . . .
```

```
            if(temp2 > p[i].ave){
               temp2 = p[i].ave;
            }
         }
         cout << "THE LOWER AVERAGE
AMONG ALL PLAYER IS : ";
         for(int i = 0 ; i < 5 ; i++)
{
            if(temp2 == p[i].ave){
               cout  <<  p[i].name  <<
endl;
               break;
            }
         }
         system ("pause");
         menu();
         break;
       case 6:
         return 0;
         break;
     }
   }

   }
```

1. What are the ways to use or declare structure in your program?

To declare a structure in a your program, you must put it above your main function. To declare one this is the syntax: struct struct_name {members};. Members is just the data type in its variable that the structure can access. You can also put another name before the semicolon.

2. How do you access a member in a structures?

To access member in a structure you just need to put a dot(.) after the struct name. For example: Struct_name.age. You will now access the data inside the member named age.

IV. ASSESSMENT

Department	Information Technology
Subject Code	CCS0007
Description	COMPUTER PROGRAMMING 2 FOR IT
Term/Academic Year	

Topic	Structures
Lab Activity No	3
Lab Activity	**Structures**
CLO	**3**

Note: The following rubrics/metrics will be used to grade students' output in the lab exercise.

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	Able to apply required data type or data	Able to identify required data type or data	Unable to identify required data type

	structure and produce correct results **(18-20pts)**	structure and produce partially correct results **(15-17pts)**	structure but does apply correctly **(12-14pts)**	**(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 outof-range data **(15-17pts)**	The program produces correct results but does not display correctly Does not check for errors and outof-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 inappropri ate 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)**

Topic	Structures
Lab Activity No	3
Lab Activity	Structures
CLO	3
Requirement Specification (30pts)	
Data type (20pts)	
Input Validation (20pts)	
Free from syntax, logic, and runtime errors (10pts)	

Delivery (10pts)	
Use of Comments (10pts)	
TOTAL	