**Instructions:**

1. Create a MS VisualStudio solution named **Assignment11**.
2. Create projects according to the assigned problems.
3. Use techniques and experience that you have obtained from previous chapters and this chapter.
4. Validate all input.
5. Code and test your programs. They **MUST** be completed without syntax, logic and run-time errors.
6. Include comments of your name, date, and brief descriptions in all source codes.  Comment precondition and postcondition on all of your user-defined functions.
7. Compress **Assignment11**folder into **ONE** zipped file.
8. Submit or re-submit your zipped file before its due date&time.

int main()  
{  
 do  
 {  
 system("cls");  
 cout << "\n\tCMPR121:  Chapter11 - Structured Data by Prof Q (3-1-2024) ";  
 cout << "\n\t" << string(80, char(205));  
 cout << "\n\t 3. Corporate Sales Data";  
 cout << "\n\t 5. Weather Statistics Modification";  
 cout << "\n\t 7. Customer Accounts";   
    cout << "\n\t 8. Search Function for Customer Accounts Program";  
 cout << "\n\t9. Speaker's Bureau";  
    cout << "\n\t10. Search Function for the Speaker's Bureau Program";  
 cout << "\n\t12. Course Grade";  
 cout << "\n\t" << string(80, char(196));  
 cout << "\n\t 0. quit";  
 cout << "\n\t" << string(80, char(205));  
  
 switch (inputInteger("\n\tOption: ", 0, 20))  
 {  
 case 0: exit(0);  
 case 3: challenge3(); break;  
 case 5: challenge5(); break;  
 case 7: challenge7(); break;  
 case 8: challenge8(); break; // 10pts Extra Credit by adding the search function on top of challenge7  
 case 9: challenge9(); break;  
 case 10: challenge10(); break; // 10pts Extra Credit by adding the search function on top of challenge9  
 case 12: challenge12(); break;   
 }  
  
 cout << "\n\n";  
 system("pause");  
 } while (true);  
  
 return 0;

3 Corporate Sales Data

Write a program that uses a structure to store the following data on a company division:

Division Name (such as East, West, North, or South)

First-Quarter Sales

Second-Quarter Sales

Third-Quarter Sales

Fourth-Quarter Sales

Total Annual Sales

Average Quarterly Sales

The program should use four variables of this structure. Each variable should represent one of the following corporate divisions: East, West, North, and South. The user should be asked for the four quarters’ sales figures for each division. Each division’s total and average sales should be calculated and stored in the appropriate member of each structure variable. These figures should then be displayed on the screen.

Input Validation: Do not accept negative numbers for any sales figures.

5-- Weather Statistics Modification

Modify the program that you wrote for Programming Challenge 4 (weather statistics) so it defines an enumerated data type with enumerators for the months (JANUARY, FEBRUARY, so on). The program should use the enumerated type to step through the elements of the array.

8--Search Function for Customer Accounts Program

Add a function to Programming Challenge 7 (Customer Accounts) that allows the user to search the structure array for a particular customer’s account. It should accept part of the customer’s name as an argument then search for an account with a name that matches it. All accounts that match should be displayed. If no account matches, a message saying so should be displayed.

9. Speakers’ Bureau

Write a program that keeps track of a speakers’ bureau. The program should use a structure to store the following data about a speaker:

Name

Telephone Number

Speaking Topic

Fee Required

The program should use an array of at least 10 structures. It should let the user enter data into the array, change the contents of any element, and display all the data stored in the array. The program should have a menu-driven user interface.

Input Validation: When the data for a new speaker is entered, be sure the user enters data for all the fields. No negative amounts should be entered for a speaker’s fee.

10 -- Search Function for the Speakers’ Bureau Program

Add a function to Programming Challenge 9 (Speakers’ Bureau) that allows the user to search for a speaker on a particular topic. It should accept a key word as an argument then search the array for a structure with that key word in the Speaking Topic field. All structures that match should be displayed. If no structure matches, a message saying so should be displayed.

12 -- Course Grade

Write a program that uses a structure to store the following data:

Member Name Description

Name Student name

Idnum Student ID number

Tests Pointer to an array of test scores

Average Average test score

Grade Course grade

The program should keep a list of test scores for a group of students. It should ask the user how many test scores there are to be and how many students there are. It should then dynamically allocate an array of structures. Each structure’s Tests member should point to a dynamically allocated array that will hold the test scores.

After the arrays have been dynamically allocated, the program should ask for the ID number and all the test scores for each student. The average test score should be calculated and stored in the average member of each structure. The course grade should be computed on the basis of the following grading scale:

Average Test Grade Course Grade

91–100 A

81–90 B

71–80 C

61–70 D

60 or below F

The course grade should then be stored in the Grade member of each structure. Once all this data is calculated, a table should be displayed on the screen listing each student’s name, ID number, average test score, and course grade.

Input Validation: Be sure all the data for each student is entered. Do not accept negative numbers for any test score.