National University of Computer and Emerging Sciences



Laboratory Exercise

Computer Programming Lab

Spring 2017

Lab # 5

(CL 103)

Department of Computer Science

|  |
| --- |
| Objectives   * STRUCTURES * ARRAY OF STRUCTURE * POINTER TO STRUCTURES * DYNAMIC MEMORY WITH STRUCTURES |

Note: Carefully read the following instructions.

1. Make a word document with the convention “ROLLNO\_SECTION\_LABNO” and put all your C++ source code in it.
2. After every question paste a screenshot of your working code below the source code in the document file.
3. At the end, when you are done with your lab tasks, make your submission on slate.

|  |
| --- |
| Problem 1: |

Create a structure called Volume that uses three variables of type Distance (from the ENGLSTRC example) to model the volume of a room. Initialize a variable of type Volume to specific dimensions, then calculate the volume it represents, and print out the result.

To calculate the volume implement a function, convert each dimension from a Distance variable to a variable of type float representing feet and fractions of a foot, and then multiply the resulting three numbers.

|  |
| --- |
| Problem 2: |

Write a program that simulates a soft drink machine. The program should use a structure that stores the following data:

Drink Name  
Drink Cost  
Number of Drinks in Machine

The program should create a dynamic array of four structures. The elements should be initialized with the following data:

**Drink Name Cost Number in Machine**Cola 75 20  
Root Beer 75 20  
Grape Soda 80 20  
Cream Soda 80 20

Each time the program runs, it should enter a loop that performs the following steps:

* A list of drinks is displayed on the screen.
* The user should be allowed to either quit the program or pick a drink.
* If the user selects a drink, he or she will next enter the amount of money that is to be inserted into the drink machine.
* The program should display the amount of change that would be returned and subtract one from the number of that drink left in the machine.
* If the user selects a drink that has sold out, a message should be displayed. *The loop then repeats.*
* When the user chooses to quit the program it should display the total amount of money the machine earned.

*Input Validation: When the user enters an amount of money, do not accept negative values, or values greater than 100.00.*

|  |
| --- |
| Problem 3: |

Write a program that declares a struct to store the data of a baseball player (player’s name, number of home runs, and number of hits). Declare an array of 5 components to store the data of 5 baseball players. Your program must contain a function to input data and a function to output data. Add functions to search the array to find the index of a specific player, and update the data of a player. Before the program terminates, give the user the option to save data in a file. Your program should be menu driven, giving the user various choices.

|  |
| --- |
| Problem 4: |

Write a program that reads students’ names followed by their test scores. The program should output each student’s name followed by the test scores and the relevant grade. It should also find and print the highest test score and the name of the students having the highest test score.

Student data should be stored in a struct variable of type studentType, which has four components: studentFName and studentLName of type string, testScore of type int (testScore is between 0 and 100), and grade of type char. Suppose that the class has 20 students. Use an array of 20 components of type studentType.

Your program must contain at least the following functions:

a. A function to read the students’ data into the array.

b. A function to assign the relevant grade to each student.

c. A function to find the highest test score.

d. A function to print the names of the students having the highest test score.

Your program must output each student’s name in this form: last name followed by a comma, followed by a space, followed by the first name; the name must be left justified. Moreover, other than declaring the variables and opening the input and output files, the function main should only be a collection of function calls.

|  |
| --- |
| Problem 5: |

Write the C++ program for processing of the student’s structure. Define the array of a structure called students including following fields:  
• “First name”  
• “Last Name”  
• “CGPA”  
You should first get the number of students from input and ask user to initialize the fields of the structure for the entered number of students. Then delete the students with the same “CPGA” and sort the list based on “CGPA” and print the final result in the screen.

|  |
| --- |
| Problem 6: |

Use the time structure from Exercise 9, and write a program that obtains two time values from the user in 12:59:59 (using a string) format, stores them in struct time variables, converts each one to seconds (type int), adds these quantities, converts the result back to hours-minutes-seconds, stores the result in a time structure, and finally displays the result in 12:59:59 format.

Note: The conversion from time format to seconds should be done in a function, a Time type variable would be passed to it, it would calculate the total number of seconds from that time to 00:00:00 and then return a structure variable.

You are done with your exercise; submit.