**CSCI 1470 – Fall 2015
Lab 12 Assignment**

**Topic:** Data Structures
**Reading:** Chapter 11

Submit all source codes (\*.cpp) at the same time via email to clarenceb@uca.edu and to tvelasco1@cub.uca.edu

***\*Note: Include the following set of comments at the top of your source code for all assignments.***

***// Student Name:***

***// Assignment #: (Example: A12-1)***

***// Lab Time: Tuesday 2:40-4:30***

***/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Title of Program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***Author: Date of Work:***

***Design: Provide an general overall description of the program***

***Input:***

***Process:***

***Output:***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/***

**Assignments:**

1. ***(Save your program as* A12-1.cpp*)*** Write a C++ program that calculates the average grade earned by a student in a Computer Science course which consists of two parts: lecture sessions and lab sessions. The program should also determine the letter grade earned by the student. The average grade earned by a student is determined by adding the student’s total points earned in both parts of the course and dividing this value by the total points that were possible for the course. The letter grade is based on the standard 10-point value scale. (A = 90 – 100, B = 80 – 89, C = 70 – 79, etc.) Also determine the highest and lowest average recorded for the course.
	1. Specify a structure named **StudentRecord** to store the following data for each student:
		1. Student Name.
		2. Total Points Earned by the student in the Lecture Sessions.
		3. Total Points Earned by the student in the Lab Sessions.
		4. Overall Total Points Earned by the student in the course.
		5. Average Grade for the student.
		6. Letter Grade Earned by the student.
	2. Specify an array of records (structures) named **student** to store the data for each student in the course. Set the maximum number of students to 50, but ***let the user specify*** how many students are in the course.
	3. Use ***main( )*** as the driver function. You do ***not*** have to make the program repeatable.
	4. Write appropriate functions that ***main( )*** calls to accomplish the following tasks:
		1. Read from the user the number of students taking the course and the total points that were possible for the course.
		2. Read the following data for each student and store the data in each **StudentRecord** structure in the array: the student name, the total points earned in the lecture sessions, total points earned in the lab sessions.
		3. Calculate the total points earned by each student, the average grade for each student, and determine the earned letter grade based on the calculated average. Store the total points earned, the average grade, and the earned letter grade in the appropriate member of the **StudentRecord** structure.
		4. Determine the highest and lowest average recorded for the course.
		5. Display the data for each student in a table format. Display all numeric values to a tenth of a decimal.

  ***Sample Input:***

Enter the number of students enrolled in the course: ? 3

 Enter the total points that are possible for the course: 1200

 Enter the information for Student 1:

 Name: ? Mary Hill

 Lecture Points Earned: 750

 Lab Points Earned: 350

Enter the information for Student 2:

 Name: ? Mike Smith

 Lecture Points Earned: 600

 Lab Points Earned: 200

Enter the information for Student 3:

 Name: ? Lisa Dobbs

 Lecture Points Earned: 725

 Lab Points Earned: 275

**Sample Output:**

Student Name Lecture Pts Lab Pts Total Pts Average Letter Grade
Mary Hill 750.0 350.0 1100.0 91.7 A

Mike Smith 600.0 200.0 800.0 66.7 D

Lisa Dobbs 725.0 275.0 1000.0 83.3 B

Highest Average: 91.7

Lowest Average: 66.7

1. ***(Save your program as* A12-2.cpp*)*** A local university would like to hire you to write a C++ program that stores the home address and current college address for their students and displays this information in a report.
	1. **Address**: Specify a structure named **Address** to store the ***street address, the city/state***, and the ***zip code***. ***You will use this structure as the data type for the required addresses in the* StudentAddress *structure.***
	2. Specify a structure named **StudentAddress** to store the following data for each student:
		1. Name
		2. Home Address. ***(Use the structure data type Address. Refer to Nested Structures.)***
		3. College Address. ***(Use the structure data type Address. Refer to Nested Structures.)***
	3. Specify a one-dimensional array to store the student address records. Set the maximum number of student records to 50 (the school can only accommodate 50 students each school year), but ***let the user specify*** how many student records need to be entered.
	4. Use ***main( )*** as your driver function. The program does ***not*** need to be repeatable.
	5. Write appropriate functions that ***main( )*** calls to accomplish the following tasks:
		1. Read the number of students in the school.
		2. Read and store the name and required addresses for each student.
		3. Display each student’s name and both addresses.

**Sample I/O:**

Number of students (must be a positive number less than or equal to 50). = ? 3

Please enter the information for student 1.

Name: Mary Jones

 Home Address:

 Street Address: 45 Ash

 City, State: Morrilton, AR

 Zip Code: 72110

 College Address:

 Street Address: 32 Elm

 City, State: Conway, AR

 Zip Code: 72034

Please enter the information for student 2.

Name: Lisa Mills

 Home Address:

 Street Address: 34 Spruce

 City, State: Vilonia, AR

 Zip Code: 72173

 College Address:

 Street Address: 56 Pine

 City, State: Conway, AR

 Zip Code: 72032

Please enter the information for student 3.

Name: Mike Hester

 Home Address:

 Street Address: 3 Oak

 City, State: Greenbrier, AR

 Zip Code: 72058

 College Address:

 Street Address: 5 Cedar

 City, State: Conway, AR

 Zip Code: 72034

The Student Address report is:

Mary Jones

 Home: 45 Ash

 Morrilton, AR 72110

 College: 32 Elm

 Conway, AR 72034

Lisa Mills

 Home: 34 Spruce

 Vilonia, AR 72173

 College: 56 Pine

 Conway, AR 72032

Mike Hester

 Home: 3 Oak

 Greenbrier, AR 72058

 College: 5 Cedar

 Conway, AR 72034

# CSCI 1470 – Fall 2015

# Lab 12 Out-of-class Assignment

**Due Date: Monday November 30, 11:00PM**

**Topic:** Data Structures
**Reading:** Chapters 11

Submit all source codes (\*.cpp) at the same time via email to clarenceb@uca.edu and to tvelasco1@cub.uca.edu

***\*Note: Include appropriate comments at the top of your source code for all assignments.***

**Assignments:**

1. ***(Save this file as* B12-1.cpp***.)* In this assignment, you will write a program that stores student information concerning the student’s scores on the ACT test. You may assume that there are less than 50 students, but the user should be allowed to specify the number of students. You should read in the student first and last name and the number of times that the student has taken the ACT test. You should assume that the maximum number of times that a student can take the ACT is 10 times. For each test, you will read in the student subscores for the four sections of the test, which are English, Math, Reading and Science.
	1. You need to create a structure named **SingleActResults** that will contain the following information:
		1. English subscore;
		2. Math subscore;
		3. Reading subscore;
		4. Science subscore.
	2. You need to use a structure named **StudentActscores** that will contain the following information:
		1. Student name;
		2. Number of times that the student has taken the ACT;
		3. Four subscores for each time that the student has taken the test. This information should be stored as array of type **SingleActResults** with length 10;
		4. Four subscores for the student’s ACT super score, which is the highest score for each subscore over all of the times that the student has taken the test. This information should be stored in a separate member of the structure and should be of type **SingleActResults**.
	3. Specify a one-dimensional array to store the student ACT scores. Set the maximum number of student records to 50, but ***let the user specify*** how many student records need to be entered.
	4. Use ***main( )*** as your driver function. The program does ***not*** need to be repeatable.
	5. Write appropriate functions that ***main( )*** calls to accomplish the following tasks:
		1. Read the number of students to process.
		2. Read and store the name and ACT scores for each student.
		3. Display each student’s name, name of subscore and student’s highest score for each subscore.
		4. Display ACT super score, which is the average of their highest scores for the subscore.

**Sample I/O:**

Number of students (must be a positive number less than or equal to 50). = ? 2

Please enter the information for student 1.

Name: Mary Jones

How many times did the student take the ACT? 2

Please enter scores for ACT Test 1:

 English: 25

 Math: 28

 Reading: 19

 Science: 22

Please enter scores for ACT Test 1:

 English: 24

 Math: 26

 Reading: 23

 Science: 25

Overall ACT super score for Mary Jones:

 English: 25

 Math: 28

 Reading: 23

 Science: 25

 ACT Super Score: 25.25

Please enter the information for student 2.

Name: Mike Hester

How many times did the student take the ACT? 3

Please enter scores for ACT Test 1:

 English: 22

 Math: 23

 Reading: 15

 Science: 18

Please enter scores for ACT Test 2:

 English: 21

 Math: 25

 Reading: 17

 Science: 19

Please enter scores for ACT Test 2:

 English: 24

 Math: 22

 Reading: 20

 Science: 21

Overall ACT super score for Mike Hester:

 English: 24

 Math: 25

 Reading: 20

 Science: 21

 ACT Super Score: 22.5