**CSCI 1470 – Fall 2015
Lab 10 Assignment**

**Topic:** Parallel Arrays
**Reading:** Chapter 7.

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: A10-1)***

***// 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:***

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

**1. *(Save this file as* A10-1.cpp***.)*

In this assignment, you will write a program that stores student information in several parallel arrays. A user of the program can input the student’s last name (properly capitalized) and whether they need the student’s email address, mailing address or phone number. If the program finds the name, the program will output either the student’s email address, the student’s mailing address, including street address, city, state and zip code, or the student’s phone number; otherwise, the program will output “Student not found”.

The following functions will be written:

int LinearSearchLastName(string SearchName, string LastNameArray[], int arraySize);

This function inputs the SearchName and the LastNameArray. It performs a linear search through the LastNameArray, comparing each name in the array with SearchName. If the name is found, it returns the integer value of the array location. If the name is not found, it returns -1, which means that the name was not found.

void OutputEmailAddress(int arrayLocation, string FirstNameArray[], string LastNameArray[], string EmailArray[], int arraySize);

Output first and last name of student, along with email address.

void OutputMailingAddress(int arrayLocation, string FirstNameArray[], string LastNameArray[], string StreetArray[], string CityArray[], string StateArray[], unsigned long int ZipCodeArray[], int arraySize);

Output first and last name of student, along with mailing address.

void OutputPhoneNumber(int arrayLocation, string FirstNameArray[], string LastNameArray[], unsigned int AreaCodeArray[], unsigned long int PhoneNumberArray[], int arraySize);

Output first and last name of student, along with phone number, formatted at (xxx) xxx-xxxx.

Please use the following data:

string FirstNameArray[] = {"Bob", "Mary", "Steve", "Juan", "David", "Laura","Jill", "Nathan", "Victoria", "Ashley"};

string LastNameArray[] = {"Martinez", "Holloway", "Johnston", "Rivera", "Bratton", "Smith", "Robeski", "Hale", "Stevens", "Graham"};

string StreetArray[] = {"543 S. Main St", "14 Commerce St. Apt 3b", "7612 Elm St.", "78 Markham Ave.", "9 Sarah Lane", "508 Willow St", "321 Red Oak Dr.", "47 Bruce St.", "760 College Ave.", "1021 S. Chester Ave"};

string CityArray[] = {"Perry", "Vilonia", "Conway", "Little Rock", "Texarkana", "Greenbrier", "Monroe", "Conway", "Jackson", "Fort Smith"};

string StateArray[] = {"AR", "AR", "AR", "AR", "TX", "AR", "LA", "AR", "MS", "AR"};

unsigned long int ZipCodeArray[] = {72126, 72173, 72034, 72207, 75503, 72058, 71207, 72032, 39207, 72906};

string EmailArray[] = {"BMartinez@aol.com", "MHolloway@cub.uca.edu", "SJohnston@cub.uca.edu", "JRivera@yahoo.com", "DBratton@cub.uca.edu", "LSmith@cub.uca.edu", "JRobeski@gmail.com", "NHale@cub.uca.edu", "VStevens@cub.uca.edu", "AGraham@yahoo.com"};

unsigned int AreaCodeArray[] = {501, 501, 501, 501, 430, 501, 318, 501, 601, 479};

unsigned long int PhoneNumberArray[] = {6739811, 3587890, 4397718, 9781553, 7765981, 5678134, 9632313, 3580013, 9769813, 8564477};

Make sure to use unsigned long int both the zip code and the phone number. Format the phone number properly; for instance, the first phone number should be displayed as (501) 673-9811

# CSCI 1470 – Fall 2015

# Lab 10 Out-of-class Assignment

**Due Date: Monday November 9, 11:00PM**

**Topic:** Parallel Arrays

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.***

**Assignments:**

**1. *(Save this file as* B10-1.cpp***.)*

In this assignment, you will write a program that stores information about student grade point average and account balances in several parallel arrays. A user of the program can input the student’s last name (properly capitalized) and whether they need the student’s grade point information, balance owed to the university for tuition and fees and balance on their BearCard for dining services. If the program finds the name, the program will output either the number of credit hours that the student has completed, along with their grade point average, the balance owed to the university for tuition and fees or the balance on the student’s BearCard; otherwise, the program will output “Student not found”.

The following functions will be written:

int LinearSearchLastName(string SearchName, string LastNameArray[], int arraySize);

This function inputs the SearchName and the LastNameArray. It performs a linear search through the LastNameArray, comparing each name in the array with SearchName. If the name is found, it returns the integer value of the array location. If the name is not found, it returns -1, which means that the name was not found.

void OutputGradeInformation(int arrayLocation, string FirstNameArray[], string LastNameArray[], unsigned short int CreditHoursCompleted[], unsigned short int QualityPointsEarned[], int arraySize);

Output first and last name of student, along with the number of credit hours completed and the student’s grade point average.

void OutputRegistrarBalance(int arrayLocation, string FirstNameArray[], string LastNameArray[], double RegistrarBalance[], int arraySize);

Output first and last name of student, along with a statement concerning how much the student owes the university.

void OutputBearCardBalance(int arrayLocation, string FirstNameArray[], string LastNameArray[], double BearCardBalance[], int arraySize);

Output first and last name of student, along with the student’s BearCard balance.

Please use the following data:

string FirstNameArray[] = {"Bob", "Mary", "Steve", "Juan", "David", "Laura","Jill", "Nathan", "Victoria", "Ashley"};

string LastNameArray[] = {"Martinez", "Holloway", "Johnston", "Rivera", "Bratton", "Smith", "Robeski", "Hale", "Stevens", "Graham"};

unsigned short int CreditHoursCompleted[] = {32, 56, 16, 28, 50, 93, 105, 73, 16, 0};

unsigned short int QualityPointsEarned[] = {103, 134, 61, 29, 155, 335, 199, 292, 56, 0};

double RegistrarBalance[] = {1203.1, 87.5, 35, 3671.81, 0, 678.40, 72, 1080.5, 81.7, 2175};

double BearCardBalance[] = {753.15, 259.35, 875, 18.23, 500, 1.35, 341.08, 79.11, 120.8, 349.25};

Make sure to use unsigned short int both the credit hours completed and the quality points earned. Format the account balances using a dollar sign $ and two decimal places after the decimal.

To calculate the grade point average (i.e., GPA), divide the quality points earned by the credit hours completed and output the answer using 2 decimal places (i.e., 3.17).