

**CSC 111E: Lab #10 – Arrays, 2-D Arrays**

Lab Date: Thursday, 11/7/2013

Due Date: Friday, 11/8/2013 @ 5:00pm

*Purpose:* The purpose of this lab is to have you gain additional experience with making use of files in your programs, to gain additional experience in using arrays, and to introduce 2-d arrays.

*Program 1: Parallel Arrays*

Make sure to read this entire description before starting any work problem solving and programming.

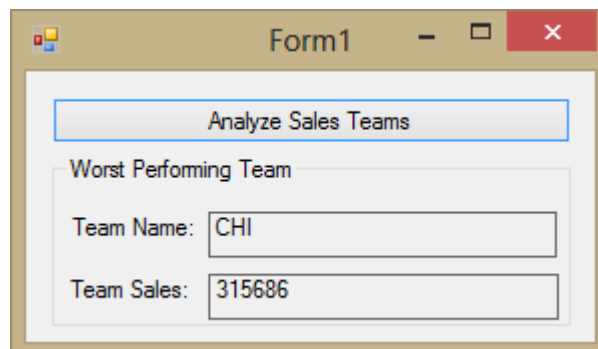
You will be given a file which contains two pieces of information about each sales team working for a company – the division name (which is based on a region of the country) and the sales generated by that team. The file is formatted as below:

```
# of teams
Team #1 name
Team #1 sales
Team #2 name
Team #2 sales
... names and sales info repeated as necessary...
```

Your task is to determine the team that had the lowest sales last year. While you could do this while just reading in the file, please do the following instead to get experience with arrays.

- Open the file
- Read in the number of teams
- Create two appropriately sized arrays (sized to the number of teams) – one holding strings (for the names) and one holding doubles (for the sales)
- Read in the name and sales data, with each line going into the appropriate array
- Close the file
- Do your minimum analysis by visiting all the entries in the array
- Print the name and sales information for the worst performing team to the GUI.

The GUI below shows the correct output given the input file you have been given.



The shell of a program, the GUI, and a data file have been created for you in *Lab10Program1.zip* available in Sakai. Below is the file will be working with, which has the name “teamsales.txt”.

4  
ATL  
422150  
NYC  
1125200  
CHI  
315686  
SOCAL  
654390

*Program 2: Daily customer analysis*

Make sure to read this entire description before starting any work problem solving and programming.

You have access to a set of data of the number of customers that visited your store in the last week. We will assume you are only open Monday-Friday from 9am to 5pm each day. The data is available for each hour of each day, meaning there are 40 pieces of data. We will consider this data as a 5x8 array (5 days, 8 hours per day) of doubles.

You will be given a file that has exactly 40 lines (5 days \* 8 hours per day). Each line will have an integer on it. The first 8 values will represent day 0 (Monday), the second 8 values will be for day 1 (Tuesday), and so on.

Your goal is to, using a 2-d array, represent the data and

- a) Report the total number of customers
- b) Report the day with the most customers

*There is a lot of verbiage in my descriptions below, but if you think through the big ideas of what should be done, the ideas should be able to be broken down into small chunks of work (remember that a lot of what you want to do is to use a For..Next loop to visit all entries in a given array).*

When the user presses the “Analyze Weekly Customers” button in the GUI,

Read in the data from the file

- Open the customer data file (“customerdata.txt”)
- Create an appropriately sized 2-d array (5 rows by 8 columns)
- Read in, using a nested loop, all of the customer data from the file. The nested loop should include an outer loop that loops from 0 to 4 (this covers the days) and an inner loop that loops from 0 to 7 (this covers the hours per day). Each line read from the file should be converted to a double and stored in the appropriate address in the 2-d array. The address is based on the variables controlling the outer and inner loops.
- Close the customer data file

Find the total number of customers

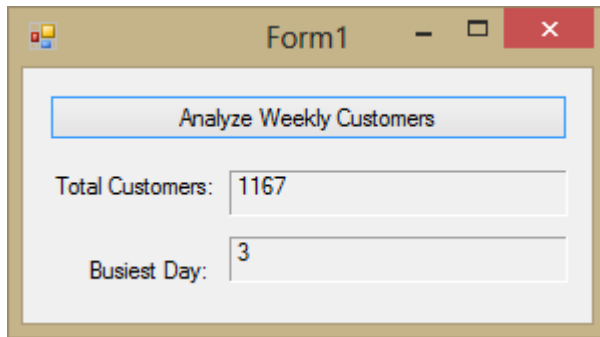
- After getting the data in from the file, we will sum the total number of customers. First, before anything else, initialize a variable representing the total number of customers to zero.
- Then create a nested loop, where the nested loop includes an outer loop that loops from 0 to 4 (this covers the days). The work for the outer loop should include:
  - Running an inner loop that loops from 0 to 7 (this covers the hours per day), with the inside of that loop adding the data being visited in the array to the running total.
- After the nested loops, print to the “Total Customers” output label the total number of customers you counted.

Find the day with the most customers

- First initialize an integer variable maxValue to -1 and an integer variable maxAddress to -1. These will help us in finding the “busiest day” (day with maximum number of customers).

- Create a nested loop that works as follows. The nested loop should include an outer loop that loops from 0 to 4 (this covers the days). The work being done within the outer loop should include:
  - Initializing a variable representing total number of customers for that day to 0;
  - Running an inner loop that loops from 0 to 7 (this covers the hours per day), with the inner loop's work being adding to the total for the given day;
  - When the inner loop finishes, comparing the discovered "day total" to the maxValue (maximum customers in a day seen so far) and updating the maximum (maxValue) and where that maximum is in the array (maxAddress) variables as needed.
- Finally, print the day number (0 – 4) stored in the maxAddress variable, which represents the day with the most customers, to the "Busiest Day" output label.

The example GUI below has the correct results for the dataset you have been provided. The basic code outline, the GUI, and the example dataset can be found on Sakai in *Lab10Program2.zip*.



#### Submission

To submit this lab for grading, do the following by Friday, 11/8, at 5:00pm:

- Zip each of the projects separately.
- Upload the projects into Sakai under the Assignments, Lab10 link.

Your grade will be based on the following rubric:

Objective	Points Available	Points Earned
Program 1: Correctly opens and closes file	5	
Program 1: Reads entire file using appropriate mechanism	5	
Program 1: Stores data correctly in TWO appropriately sized arrays	12	
Program 1: Correctly computes min of data from array	12	
Program 1: Correctly shows results in GUI	5	
Program 2: Correctly opens and closes file	5	
Program 2: Reads entire files using appropriate mechanism	5	
Program 2: Stores client data correctly in appropriately sized 2-DIMENSIONAL array	15	
Program 2: Correctly computes total number of customers	15	
Program 2: Correctly computes busiest day	15	
Program 2: Correctly shows results in GUI	5	