**Student Lab Activity**



1. Lab # CIS CIS170C-A5
2. Lab 5 of 7: Arrays and Strings
3. Lab Overview - Scenario/Summary

You will code, build, and execute two programs requiring arrays and strings.   
  
First program *(Video Game Player Program)*: Determine the average score for a group of players and then determine who scored below average.

Second program *(Pig Latin)*: Convert words in a phrase to pig latin.

Learning outcomes:

1. Be able to explain the need for arrays in a program.
2. Be able to determine the appropriate array data type to use in a given program.
3. Be able to write a program that implements arrays.
4. Be able to explain the way memory is allocated for arrays in a program.
5. Be able to explain the fact that arrays are objects in C++.
6. Be able to write a program that implements strings.
7. Deliverables

|  |  |  |
| --- | --- | --- |
| **Section** | **Deliverable** | **Points** |
| **Part A** | Step 5: Program Listing and Output | **20** |
| **Part B** | Step 5: Program Listing and Output | **25** |

1. Lab Steps

Preparation:

If you are using the Citrix remote lab, follow the login instructions located on the iLab tab in Course Home.

Locate the Visual Studio 2010 icon and launch the application.

Lab:

|  |
| --- |
| **Part A: Video Game Player Program** |
| **Step 1:** Requirements |
| Write a program to do the following:  In main, declare a *PlayerName Array* and a *Score Array*. Declare the size of the arrays to be 100.   In the *InputData function*, input the player name and score into the arrays for an unknown number of players up to 100.  In the *DisplayPlayerData function*, display the name and score of each player.  In the *CalculateAverageScore function*, calculate the average score and return it by value.  In the *DisplayBelowAverage function*, display the name and score for any player who scored below the average. Do not use global variables.  Output from Program:  **Enter Player Name (Q to quit): Bob**  **Enter score for Bob: 3245**  **Enter Player Name (Q to quit): Sue**  **Enter score for Sue: 1098**  **Enter Player Name (Q to quit): Dave**  **Enter score for Dave: 8219**  **Enter Player Name (Q to quit): Pat**  **Enter score for Pat: 3217**  **Enter Player Name (Q to quit): Q**  **Name Score**  **Bob 3245**  **Sue 1098**  **Dave 8219**  **Pat 3217**  **Average Score: 3944.75**  **Players who scored below average**  **Name Score**  **Bob 3245**  **Sue 1098**  **Pat 3217**  **Press any key to continue . . .** |
| **Step 2:** Processing Logic |
| Using the pseudocode below, write the code that will meet the requirements.  Main Function  Declare the player name and score arrays, number of players, and average score.  Call the InputData function  Call the DisplayPlayerData function  Call the CalculateAverageScore function and assign the returned value in average score  Call the DisplayBelowAverage function  InputData function  While the number of players is less than the length of the array  Prompt for the player's name  If the user entered Q, break out of the loop  Prompt the user for the player's score  Add 1 to the number of players  End-While  DisplayPlayerData function  Display the name and score of each player  CalculateAverageScore function  Add up the scores and divide by the number of scores to calculate the average score  Display the average score  Return the average score to main  DisplayBelowAverage function  Display the names and scores of all players who scored below the average score |
| **Step 3:** Create a New Project |
| Create a new project and name it LAB5A. Write your code using the Processing Logic in Part A, Step 2. Make sure you save your program. |
| **Step 4:** Compile and Execute |
| 1. Compile your program. Eliminate all syntax errors. 2. Build your program and verify the results of the program. Make corrections to the program logic if necessary until the results of the program execution are what you expect. |
| **Step 5:** Print Screen Shots and Program |
| 1. Capture a screen print of your output. (Do a PRINT SCREEN and paste into an MS Word document.) 2. Copy your code and paste it into the same MS Word document that contains the screen print of your output. 3. Save the Word document as Lab05A\_LastName\_FirstInitial. |
| **END OF PART A** |

|  |
| --- |
| **Part B: Pig Latin** |
| **Step 1:** Requirements |
| Write a program that will input a phrase and convert it to pig latin. Put each word in a separate element of a string array. Remove the first letter from each word and concatenate it to the end of the word followed by “ay.”  Sample Output from Program:  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **\* You will be prompted to enter a string of \***  **\* words. The string will be converted into \***  **\* Pig Latin and the results displayed. \***  **\* Enter as many strings as you would like. \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **Enter a group of words or ENTER to quit: Computer Programming is fun to learn!**  **Original words: Computer Programming is fun to learn!**  **New Words: omputercay ogrammingpray isway unfay otay earnlay!**  **Enter a group of words or ENTER to quit: Quit**  **Pig Latin Hint:** If a word begins with one or more consonants, move the consonant or consonant cluster to the end of the word. Add the letters "ay" to the end of the word. So, "pig" would be "igpay," and “latin” would be “atinlay.” |
| **Step 2:** Processing Logic |
| Using the pseudocode below, write the code that will meet the requirements.  Main function  Display the heading  While the condition is true  Prompt the user for group of words or Enter to quit  Display original words  Call function pigLatinString( )  End while  pigLatinString( ) function  Declare and initialize string variables len, counter, start, begin, word and newString  While condition is true  Call find() and pass a space and start as parameters and return the returned value  to start  if start equals to string::npos  jump outside the loop permanently  call substr() function  display the word  update newString  increment start by one  assign start to begin  End While  Call substr()  Update newString  Return newString |
| **Step 3:** Create a New Project |
| Create a new project and name it LAB5B. Write your code using the Processing Logic in Part B, Step 2. Make sure you save your program. |
| **Step 4:** Compile and Execute |
| 1. Compile your program. Eliminate all syntax errors. 2. Build your program and verify the results of the program. Make corrections to the program logic if necessary until the results of the program execution are what you expect. |
| **Step 5:** Print Screen Shots and Program |
| 1. Capture a screen print of your output. (Do a PRINT SCREEN and paste into an MS Word document.) 2. Copy your code and paste it into the same MS Word document that contains the screen print of your output. 3. Save the Word document as Lab05B\_LastName\_FirstInitial. |
| **END OF LAB** |