**Renee Thomas**

**CIS170C\_Lab05A**

**2/10/14**

**Lab # CIS CIS170C-A5B**

// ---------------------------------------------------------------

// Programming Assignment: LAB5B

// Developer: Renee Thomas

// Date Written: 2/10/14

// Purpose: Arrays - Strings

// ---------------------------------------------------------------

# include <iostream>

# include <iomanip>

# include <string>

# include <cctype>

using namespace std;

string pigLatinString(string eStr);

void main()

{

string str;

cout<<setw(50)<<left<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\* You will be prompted to enter a string of \*"<<endl;

cout<<"\* words. The string will be converted into \*"<<endl;

cout<<"\* Pig Latin and the results displayed. \*"<<endl;

cout<<"\* Enter as many strings as you would like. \*"<<endl;

cout<< "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

while(true)

{

cout << "\nEnter a group of words or ENTER to quit: ";

getline(cin, str);// use getline function to acquire all of string including spaces

if(str.length() == 0) // if user just hits enter, break out of loop

{

break;

}

cout << "Original words: " << str << endl;// output string that user input

cout << "New words: " << pigLatinString(str) << endl; // output piglatin version of string using the pigLatinString(str) function

}

}

string pigLatinString(string eStr)

{

// initialize variables

int start = 0;

int begin = 0;

string word, newString = "";

while(true) // while there is still string left loop through the string

{

start = eStr.find(' ', start); // look for the next space starting from "start"

word = eStr.substr(begin, start - begin); // output the string from begin starting from start minus begin (if start is 18 and begin is 5 you would print out the five letter word starting from position 13)

// create the new string by appending each updated word to the last version of word

// word.substr(1) takes all letters minus the first letter of word

// word.substr(0,1) grabs the first letter from word

newString += word.substr(1) + word.substr(0,1) + "ay ";

if (start == string::npos) // check to see if at the end of eStr: if so break out of loop

break;

start ++; // add 1 to start

begin = start; // make begin equal start to start looking for next space/ word

}

return newString;

}

