Exercise 7-1: Simple Data Types Do More than Just Track Inventory

#include <iostream>

#include <iomanip> //can be used for formatting in later program modding...

#include <string>

#include <fstream> //included for opening, reading, writing, and closing file...

using namespace std;

bool reimburseChecker(char character)

{

switch (character)

{

case 'C':

case 'D':

case 'H':

return true;

default:

return false;

}

}

int main()

{

ifstream inFile; //included for writing to file...

ofstream outFile; //included for reading file...

const int SIZE = 11; //size can be changed and txt edited...

string codes[SIZE];

string items[SIZE];

double prices[SIZE];

char character;

bool status;

inFile.open("pharmdat.txt"); //opens txt file for input – writing...

outFile.open("pharmdat.out"); //opens out file for output – reading...

cout << "Creating file with pharmacy data info...\n"; //message displayed when program is run...

outFile << "Pharmacy Data...\n";

for (int i = 0; i < SIZE; i++)

{

inFile >> codes[i] >> items[i] >> prices[i];

}

// OUTPUTS...

for (int i = 0; i < SIZE; i++)

{

character = codes[i][0]; // 0 is the first character of string array[i]...

outFile << codes[i] << " " << items[i] << " ";

status = reimburseChecker(character); // SEARCH FILE...

if (status == true)

{

outFile << "Qualified Expense "; // reimbursement output...

}

outFile << std::fixed << std::setprecision(2) << prices[i] << " \n";

}

inFile.close(); //closes txt file

outFile.close(); //closes out file

system("pause");

return 0;

}

Exercise 7-2: Search and Replace

(with user input)

#include <iostream>

#include <string>

using namespace std;

// REPLACE WORD FUNCTION...

void ReplaceWord(const int SIZE, string phrase[], string replaceWord, string newWord)

{

for (int i = 0; i < SIZE; i++)

{

if (phrase[i] == replaceWord)

{

phrase[i] = newWord;

}

// REPLACE WORD IF FOLLOWED BY PUNCTUATION...

if (phrase[i] == replaceWord + ',')

{

phrase[i] = newWord + ',';

}

if (phrase[i] == replaceWord + '.')

{

phrase[i] = newWord + '.';

}

if (phrase[i] == replaceWord + '!')

{

phrase[i] = newWord + '!';

}

if (phrase[i] == replaceWord + '?')

{

phrase[i] = newWord + '?';

}

}

}

int main()

{

const int SIZE = 10; //size can be changed...

string phrase[SIZE];

string replaceWord;

string newWord;

// DETERMINE PHRASE...

cout << "Enter a 10 word phrase: "; //word limit can be changed...

for (int i = 0; i < SIZE; i++)

{

cin >> phrase[i];

}

cout << "Word to replace: ";

cin >> replaceWord;

cout << "New word: ";

cin >> newWord;

// REPLACE WORD...

ReplaceWord(SIZE, phrase, replaceWord, newWord); //replace word function...

// OUTPUT RESULT...

cout << "Result: ";

for (int i = 0; i < SIZE; i++)

{

cout << phrase[i] << " ";

}

cout << endl << endl;

system("pause");

return 0;

}

Exercise 7-3: Applied Search and Replace

(with punctuation accounted for)

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

// REPLACE WORD FUNCTION...

void ReplaceWords(const int SIZE, string phrase[], string replaceWord[], string newWord[])

{

for (int j = 0; j < 5; j++)

{

for (int i = 0; i < SIZE; i++)

{

if (phrase[i] == replaceWord[j])

{

phrase[i] = newWord[j];

}

// REPLACE WORD IF FOLLOWED BY PUNCTUATION...

if (phrase[i] == replaceWord[j] + ',')

{

phrase[i] = newWord[j] + ',';

}

if (phrase[i] == replaceWord[j] + '.')

{

phrase[i] = newWord[j] + '.';

}

if (phrase[i] == replaceWord[j] + '!')

{

phrase[i] = newWord[j] + '!';

}

if (phrase[i] == replaceWord[j] + '?')

{

phrase[i] = newWord[j] + '?';

}

}

}

}

int main()

{

ifstream inFile; //included for writing to file...

ofstream outFile; //included for reading file...

const int SIZE = 92; //size can be changed and txt file edited...

string phrase[SIZE];

string replaceWord[5] = { "apparant", "greatful", "medievel", "seperate", "wierd" };

string newWord[5] = { "apparent", "grateful", "medieval", "separate", "weird" };

inFile.open("sentences.txt"); //opens txt file for input - writing...

outFile.open("sentences.out"); //opens out file for output - reading...

// MESSAGE DISPLAYED WHEN PROGRAM IS RUN...

cout << "Creating file with sentences...\n";

// COPY TXT DATA TO ARRAY...

for (int i = 0; i < SIZE; i++)

{

inFile >> phrase[i];

}

// REPLACE WORDS...

ReplaceWords(SIZE, phrase, replaceWord, newWord);

// OUTPUT RESULT...

outFile << "\*Result\*" << endl << endl; //message displayed at beginning of OUT file...

for (int i = 0; i < SIZE; i++)

{

outFile << phrase[i] << " ";

if (i % 15 == 0 && i != 0)

{

outFile << endl;

}

}

outFile << endl << endl;

inFile.close(); //closes txt file...

outFile.close(); //closes out file...

system("pause");

return 0;

}