Lab 10

Source Code

// Preston Knibbe

#include <iostream>

#include <fstream>

#include <vector>

#include <stdlib.h>

#include <time.h>

using namespace std;

class Dictionary {

public:

Dictionary();

Dictionary(string);

string wordChooser(int, int);

private:

vector<string> wordBank;

string fileName;

void readFile();

void printStats();

int wordCount();

int characterCount();

double wordLength();

void alphaWordCount();

string longestWord();

};

Dictionary::Dictionary() {

cout << "Filename to read: " << endl;

cin >> fileName;

readFile();

//printStats();

}

Dictionary::Dictionary(string file) {

fileName = file;

readFile();

//printStats();

}

void Dictionary::readFile() {

fstream wordList(fileName.c\_str());

if (!wordList) {

cout << "Error in reading file" << endl;

}

string temp;

while (wordList >> temp) {

wordBank.push\_back(temp);

}

}

string Dictionary::wordChooser(int startRange, int endRange) {

srand(time(0));

vector<string> tempList;

for (int a = 0; a < wordBank.size(); a++) {

if (wordBank[a].length() > startRange &&

wordBank[a].length() < endRange) {

tempList.push\_back(wordBank[a]);

}

}

int random = rand() % tempList.size() + 0;

return tempList[random];

}

int Dictionary::wordCount() {

return wordBank.size();

}

int Dictionary::characterCount() {

int charCount = 0;

string temp;

for (int a = 0; a < wordBank.size(); a++) {

temp = wordBank[a];

for (int b = 0; b < temp.length(); b++) {

charCount++;

}

}

return charCount;

}

double Dictionary::wordLength() {

double avg = characterCount() / wordCount();

return avg;

}

void Dictionary::alphaWordCount() {

cout << "Words in dictionary by letter" << endl;

for (int a = 97; a <= 122; a++) {

cout << (char) a << ": ";

int counter;

for (int b = 0; b < wordBank.size(); b++) {

if (wordBank[b].at(0) == (char) a) {

counter++;

}

}

for (int c = 0; c <= (counter/2000); c++) {

cout << "\*";

}

cout << endl;

}

}

string Dictionary::longestWord() {

string longest;

int length = 0;

for (int a = 0; a < wordBank.size(); a++) {

if (wordBank[a].length() > length) {

longest = wordBank[a];

}

}

return longest;

}

void Dictionary::printStats() {

cout << "Total number of words: " << wordCount() << endl;

cout << "\n";

cout << "Total characters: " << characterCount() << endl;

cout << "\n";

cout << "Average characters per word: "

<< wordLength()

<< endl;

alphaWordCount();

cout << "\n";

cout << "Longest word: " << longestWord() << endl;

}

class Hangman {

public:

Hangman();

void gameDriver();

void setWord(string);

private:

const int STRIKES = 6;

int strikeCounter;

string secretWord;

string missedLetters;

char guess;

string correctLetters;

};

Hangman::Hangman() {

strikeCounter = 0;

missedLetters = "";

correctLetters = "";

}

void Hangman::setWord(string word) {

secretWord = word;

}

void Hangman::gameDriver() {

cout << "Welcome to Hangman" << endl;

while(strikeCounter < STRIKES) {

char tempLetter;

cout << "Guess:";

for (int a = 0; a < secretWord.length(); a++) {

bool tester = false;

for (int c = 0; c < correctLetters.length(); c++) {

if (secretWord[a]==correctLetters[c]) {

tempLetter = correctLetters[c];

tester = true;

}

}

if (tester) {

cout << tempLetter;

} else {

cout << "\_";

}

}

cout << " misses remaining: "

<< (STRIKES - strikeCounter)

<< " missed letters: "

<< missedLetters

<< endl;

cin >> guess;

bool guessTest = false;

for (int b = 0; b < secretWord.length(); b++) {

if (guess == secretWord.at(b)) {

guessTest = true;

}

}

if (guessTest) {

correctLetters += guess;

} else {

missedLetters += guess;

strikeCounter++;

}

if (correctLetters.length() == secretWord.length()) {

break;

}

}

if (strikeCounter < STRIKES) {

cout << "You won!" << endl;

} else {

cout << "You lose, the word was " << secretWord << endl;

}

}

int main()

{

Dictionary Test("dictionary.txt");

bool playAgain = true;

do {

string word = Test.wordChooser(4,8);

Hangman Game1;

Game1.setWord(word);

Game1.gameDriver();

char play;

cout << "Do you want to play again? (Y/N)" << endl;

cin >> play;

if (play=='N') {

playAgain = false;

}

} while (playAgain);

return 0;

}

Dictionary Output (2 different dictionaries)

Total number of words: 109582

Total characters: 935171

Average characters per word: 8

Words in dictionary by letter

a: \*\*\*\*

b: \*\*\*\*\*\*\*

c: \*\*\*\*\*\*\*\*\*\*\*\*

d: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

e: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

f: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

g: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

h: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

i: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

j: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

k: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

l: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

m: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

n: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

o: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

p: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

q: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

r: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

s: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

t: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

u: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

v: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

w: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

y: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

z: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Longest word: zyzzyvas

Process returned 0 (0x0) execution time : 0.769 s

Press any key to continue.

Total number of words: 235886

Total characters: 2257223

Average characters per word: 9

Words in dictionary by letter

a: \*\*\*\*\*\*\*\*

b: \*\*\*\*\*\*\*\*\*\*\*\*\*

c: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

d: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

e: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

f: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

g: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

h: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

i: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

j: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

k: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

l: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

m: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

n: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

o: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

p: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

q: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

r: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

s: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

t: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

u: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

v: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

w: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

y: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

z: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Longest word: Zyzzogeton

Process returned 0 (0x0) execution time : 1.519 s

Press any key to continue.

Hangman Output

Welcome to Hangman

Guess:\_\_\_\_\_\_\_ misses remaining: 6 missed letters:

a

Guess:\_\_a\_\_\_\_ misses remaining: 6 missed letters:

e

Guess:\_\_a\_e\_\_ misses remaining: 6 missed letters:

i

Guess:\_\_a\_e\_\_ misses remaining: 5 missed letters: i

o

Guess:\_\_a\_e\_\_ misses remaining: 4 missed letters: io

u

Guess:\_\_a\_e\_\_ misses remaining: 3 missed letters: iou

b

Guess:\_\_a\_e\_\_ misses remaining: 2 missed letters: ioub

c

Guess:\_\_a\_e\_\_ misses remaining: 1 missed letters: ioubc

r

Guess:\_\_a\_er\_ misses remaining: 1 missed letters: ioubc

t

You lose, the word was flamers

Do you want to play again? (Y/N)

Y

Welcome to Hangman

Guess:\_\_\_\_\_\_ misses remaining: 6 missed letters:

a

Guess:\_\_a\_\_\_ misses remaining: 6 missed letters:

e

Guess:\_\_a\_e\_ misses remaining: 6 missed letters:

i

Guess:\_\_a\_e\_ misses remaining: 5 missed letters: i

r

Guess:\_\_a\_e\_ misses remaining: 4 missed letters: ir

j

Guess:\_\_a\_e\_ misses remaining: 3 missed letters: irj

h

Guess:\_\_a\_e\_ misses remaining: 2 missed letters: irjh

w

Guess:\_\_a\_e\_ misses remaining: 1 missed letters: irjhw

o

You lose, the word was flaxen

Do you want to play again? (Y/N)

Y

Welcome to Hangman

Guess:\_\_\_\_\_ misses remaining: 6 missed letters:

a

Guess:\_\_\_\_\_ misses remaining: 5 missed letters: a

e

Guess:\_\_\_\_e misses remaining: 5 missed letters: a

i

Guess:\_\_i\_e misses remaining: 5 missed letters: a

o

Guess:\_\_i\_e misses remaining: 4 missed letters: ao

y

Guess:\_\_i\_e misses remaining: 3 missed letters: aoy

f

Guess:f\_i\_e misses remaining: 3 missed letters: aoy

x

Guess:f\_i\_e misses remaining: 2 missed letters: aoyx

l

Guess:fli\_e misses remaining: 2 missed letters: aoyx

t

You won!

Do you want to play again? (Y/N)

N

Process returned 0 (0x0) execution time : 177.566 s

Press any key to continue.