#include<iostream>

#include<string>

#include<vector>

#include<ctime>

#include<fstream>

#include<cstdlib>

using namespace std;

//using this will count the numbers of inputs the user has entered to guess

//increments in the function to sotre into the following element of the arrray

int USED = 0;

class Hangman{

private:

//stores the word the user has to guess to this string

string SecretWord;

// converts the secret word to ? to display the word to user

// to guess taking single char inputs

string DisguisedWord;

//saves their guess to this string but takes the 0 element works same as char

string Aguess;

//This array is where it stores the words that are read from a file

string WordLibrary[25];

//counts the number of total guesses

int GuessCount;

//counts when the user enter a incorrect guess

int WrongGuess;

//This is for the menu when picking a category

int PickCategory;

// Not to take the spaces in the word in the set amount of guesses for the user

// it will only count the all the letters in word to set the number of guesses

// hence a space is not a letter

int NotALetter;

public:

//This array stores the guesses the user inputs

string GuessHistory[26];

string cpySecretWord;

//this is a default constructor

Hangman(){

SecretWord = "Hello";

DisguisedWord = "?????";

Aguess = "";

GuessCount = 0;

WrongGuess = 0;

PickCategory = 0;

NotALetter = 0;

}

//Acessors to access the private variables of the class members

string getSecertWord(){

return SecretWord;

};

string getDisguisedWord(){

return DisguisedWord;

};

int getGuessCount(){

return GuessCount;

};

int getWrongGuess(){

return WrongGuess;

}

int getNotAL(){

return NotALetter;

}

//functions

//This function creates convert the secretword and disguise from it the user

void setDisguisedWord(string x);

//displays what the user has done so far such as what words have been gussed so far

//also total guesses and incorrect guess made this far each time the user enters an input

void displayInfo(){

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "\nTesting Copy of SecretWord: " << cpySecretWord << endl;

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << getDisguisedWord() << endl;

cout << "\nGuess made so far " << getGuessCount();

cout << "\nInncorrect guesses so far " << getWrongGuess() << endl;

cout << "Gussed letter so far.." << endl;

//For loop to acess the Guess histroy and display all guesses made beforehand

for (int i = 0; i <= USED; i++){

cout << GuessHistory[i] << ", ";

}

};

//gives the user a option to pick from a selection provided and program

//will adjust itself depending what the user picks

void Menu();

//This function takes input from the user to verify if the gussed word is in the secret word

void MakeGuess(string& x);

//isFound bool function searches the secretword with the guessed word

//to check if the word is found within the secret word which is true else is false

bool &IsFound();

// this reads words from a file and stores them to string array WordLibrary

bool ReadAFile();

};

int main(){

// created an object of the clas and created a string guess to store input

//which is the guess the user enters

Hangman g1; string guess; int Guesses;

//title of the game

cout << "\n Hello, Welcome to HangMan!" << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

// object from the class calls readfile to obtain the word library

g1.ReadAFile();

//after reading words goes to the menu for the user to select a category

g1.Menu();

//displays the secretword for debugging

cout << "\nThis is the secret Word: " << g1.getSecertWord() << endl;

// subtracks the spaces from the lenth because spaces are not letters

//saves the total into a variable and sets it as the limit of number of guesses

Guesses = g1.getSecertWord().length();

cout << "\nLength of SecretWord including spaces " << Guesses << endl;

Guesses = Guesses - g1.getNotAL();

cout << "\nAfter excluding spaces: " << Guesses << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "\nHere is the word you must attempt to guess! " << g1.getDisguisedWord() << endl;

cout << "\nPlese enter a letter to guess " << endl;

cin >> guess;

//obtains guess from the user and runs function from the class

g1.MakeGuess(guess);

g1.displayInfo();

//enters loop until the word is completed

for (int i = 0; i < Guesses; i++){

cout << "\nPlease enter a letter to guess!" << endl;

cin >> guess;

g1.MakeGuess(guess);

g1.displayInfo();

if (g1.getDisguisedWord().compare(g1.getSecertWord()) == 0){

cout << "\ncongratulations on completing the Word!" << endl;

cout << "(>'-')> Awesome job!" << endl;

break;

}

}

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "\nThe word was: " << g1.getSecertWord() << endl;

cout << "\n\_\_\_\_End of the program!\_\_\_\_\_" << endl;

system("pause");

return 0;

}

//The menu for the user to pick a category then sets the secret word

//picked from the library of words by a random function

void Hangman::Menu(){

int word;

srand(time(NULL));

cout << "\n\*Choose a category!\*" << endl;

cout << " 1.Place \n"

<< " 2.T.V. Show\n"

<< " 3.An Animal\n"

<< " 4.Celebrity\n"

<< " 5.Random\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n" <<

"\nPlease pick a number next to the categories!\n" << endl;

cin >> PickCategory;

//checks users input!

while (PickCategory < 1 || PickCategory > 5){

cout << "\nPlease re-enter your choice! with the following corresponding options" << endl;

cin >> PickCategory;

}

// needs to call a random string array with 10 words within each array!

switch (PickCategory){

case 1: cout << "\nYou picked places!\n";

word = rand() % 5;

SecretWord = WordLibrary[word]; setDisguisedWord(SecretWord);

break;

case 2: cout << "\nYou picked T.V. Shows\n";

word = rand() % 5 + 5;

SecretWord = WordLibrary[word]; setDisguisedWord(SecretWord);

break;

case 3: cout << "\nYou picked Animals\n";

word = rand() % 5 + 10;

SecretWord = WordLibrary[word]; setDisguisedWord(SecretWord);

break;

case 4: cout << "\nYou picked celebrity\n" << endl;

word = rand() % 5 + 15;

SecretWord = WordLibrary[word]; setDisguisedWord(SecretWord);

break;

case 5: cout << "\nYou picked Random\n" << endl;

word = rand() % 5 + 20;

SecretWord = WordLibrary[word]; setDisguisedWord(SecretWord);

break;

default: cout << "\nHHAHAH IMPOSSIBLE!\n" << endl;

}

}

// replaces the secret word with symbols and spaces depending no the word

void Hangman::setDisguisedWord(string x){

cpySecretWord= x;

DisguisedWord = x;

for (int i = 0; i < SecretWord.length(); i++){

if (SecretWord[i] == ' '){

DisguisedWord[i] = ' ';

NotALetter++;

}

else{

DisguisedWord[i] = '?';

}

}

cout << "\nTesting Copy of SecretWord: " << cpySecretWord << endl;

}

//searches the secret word with the input to search if secret == input then return true!

bool &Hangman::IsFound(){

string b = Aguess;

bool c;

for (int i = 0; i < SecretWord.length(); i++){

if (cpySecretWord[i] == b[0]){

GuessCount++;

c = true;

return c;

}

}

GuessCount++;

WrongGuess++;

c = false;

return c;

}

//functon that takes input string from the user

//the uses the boolean function isFound to see an element matches the input

// if it does then sets that letter in the disguised word until the user

// solves the word provided by the program

void Hangman::MakeGuess(string& x){

Aguess = tolower(x[0]);

//saves the input fromt he user into the GuessHistory array

GuessHistory[USED] = Aguess;

USED++;

cout << "Is this your guess " << Aguess << endl;

if (&Hangman::IsFound()){

for (int i = 0; i < SecretWord.length(); i++){

if (cpySecretWord[i] == Aguess[0]){

DisguisedWord[i] = Aguess[0];

cpySecretWord[i] = ' '; //deletes that matching input to avoid repition

}

}

}

}

//reads from a file and stores those words into string array so the menu can choose from

// the selections of different words

bool Hangman::ReadAFile(){

ifstream Words("WordL.txt");

if (Words.fail()){

cout << "Fail to open file " << endl;

return false;

}

cout << "File was succeessfuly read!" << endl;

int i = 0;

while (!Words.eof()){

getline(Words, WordLibrary[i]);

cout << "Testing: " << WordLibrary[i] << endl;

i++;

}

return true;

}

**The Words in the txt file “WordL” are the following**:  
rome

the great wall of china

mount everest

eiffel tower

supreme court

the walking dead

the game of thrones

dragon ball z

adventure time

house

honey badger

donkey

shirek

chewbacca

sloth

shia lebouf

bear grylls

ronald reagan

beyonce

mila kunis

star fruit

animal crackers

obama care

tweed

peppermint