UNIVERSITY OF THE WEST INDIES CAVEHILL CAMPUS

COMP1205 Assignment

Word Guessing Game

DUE : December 3rd 2023 Midnight

In this assignment we are developing a word guessing game. You can watch the video at https://youtu.be/dsruh3cNWZg to see the functionality of the game.

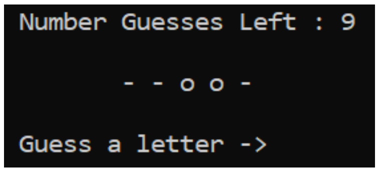
You would be given a text file that has a number of words within it, when the program is executed one of the words from the file is randomly selected. This word is not seen by the user; however each letter in the word is shown as a dash on the screen, with the exception of at least two random letters in the word.

The user would be given a number of chances to guess the letters of the word. The number of chances will be 4 more than the number of letters in the word.

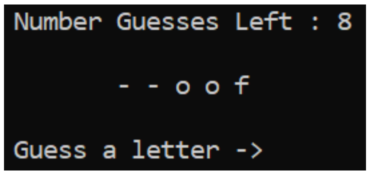
If the user guesses a letter in the word all places where that letter is in the word will no longer show as a dash but rather it would show the actual letter. If the guessed letter is not in the word, then the number of guesses left would be reduced by 1 and the user is allowed to guess again.

The round will be over if the number of guesses reaches zero or all the letters in the word are shown. At the end of each round the user would be given a message indicating that a * would end the game or a # would start a new round.

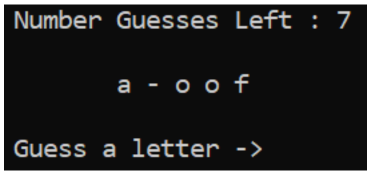
Sample round 1:



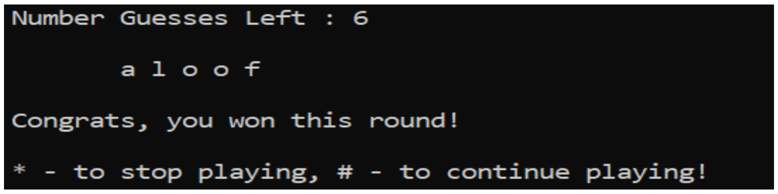
```
Number Guesses Left : 9

       - - o o -

Guess a letter ->
```



```
Number Guesses Left : 8

       - - o o f

Guess a letter ->
```

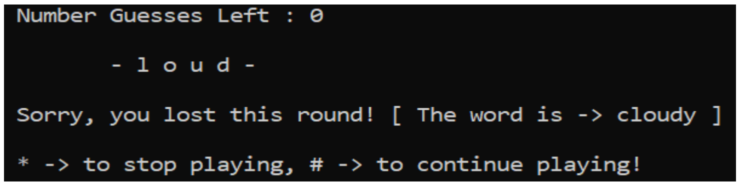


```
Number Guesses Left : 7

       a - o o f

Guess a letter ->
```



```
Number Guesses Left : 6

       a l o o f

Congrats, you won this round!

* - to stop playing, # - to continue playing!
```

Sample round 2:



```
Number Guesses Left : 0

      - l o u d -

Sorry, you lost this round! [ The word is -> cloudy ]

* -> to stop playing, # -> to continue playing!
```

The following program has some of the concepts that can help with the development of this assignment. You can run this code to see what it does;

```
#include <iostream>
#include <vector>
using namespace std;
class Letter {
      public:
      char value;
      bool visible;
      Letter(char val) : value(val), visible(true) {}
};

class WordApp {
private:
      vector<Letter> theword;
public:
      WordApp(string word) {
            for (char c : word) {
                  Letter L(c);
                  theword.push_back(L);
            }
      };
      void hide(char character) {
            for (Letter &l : theword) {
                  if (l.value == character) {
                        l.visible = false;
                  }
            }
      }
      void display() {
            for (Letter l : theword) {
                  char v = (l.visible) ? l.value : '-';
                  cout << v << " ";
            }
      };
};
int main() {
      WordApp obj("Barbados");
      obj.hide('a');
      obj.display();
      return 0;
}
```

Code Explanation:

The **Letter** class holds the individual letters of the specified word, it also has a **visible** data member which determines if the letter should be displayed to the screen or if a dash would be shown instead. When an object is created from this class the constructor accepts the letter and assigns it to the **value** data member and the **visible** data member is set to true initially.

The **WordApp** class has a **constructor**, one data member (**theword)** which is a **vector** of **Letters** and two void functions **hide()** and **display()**. The data member (**theword)** is a vector that holds in a series of **Letter** objects which **value** data member spells out the chosen word.

The **constructor** accepts a string which is a word that is supplied to the object. The for-each loop in the constructor scans through each character of the word and creates a **Letter** object from that character, and inserts that **Letter** object into the vector data member (**theword)**. When the constructor is completed the **theword** data member will have a **Letter** object for each of the characters in the supplied string.

The **hide** function accepts a character. The for-each loop in this function scans through each **Letter** object in the vector and if the **value** in the **Letter** object equals the character passed into this function, then that **Letter** object **visible** data member is set to **false**. Notice the use of the **&**, this is critical in order to create a reference to the actual **Letter** variable in the vector and not a copy of that **Letter** variable. An alternative to using a for-each loop and the **&** would be to use a regular for loop and access the vector directly via the **[]**.

The **display** function uses a for-each loop to scan through each **Letter** objects in the vector and if the **visible** data member is **true** print the **value** data member and if not print a dash.

The **main** function creates a **WordApp** object passing a string to the constructor, then call the **hide** function passing to it the character to be hidden, then call the **display** function to display the word with the specified character replaced with a dash.

Marking Scheme:

1.	Use of a Letter class (from above code)	5 Marks
2.	Use of a WordApp class (from above code)	5 Marks
3.	Use of a vector of Letter objects	5 Marks
4.	* to end game, # to play a next round	5 Marks
5.	Round won or lost messages	5 Marks
6.	Getting random word from file	5 Marks
7.	Use of h file for declaration of classes and functions	5 Marks
8.	Use of cpp file for definition of classes and functions	5 Marks
9.	Use of separate cpp file with main function	5 Marks
10.	Initial display of at least 2 characters in chosen word	5 Marks
11.	On screen indication of the number of guesses left	5 Marks
12.	Show the correct word if user did not get it right	5 Marks
13.	Good code formatting with useful comments	5 Marks

Deliverables:

1. Two **.cpp** files, one **.h** file and the **.exe** file.

Notice:

- You should not modify the **Letter** class; however the **WordApp** class would need to be modified.
- Note, here I am demonstrating a **hide()** function but it should be easy to see that you could have produce a **show()** function just as well.
- This is an individual project and hence any signs of cheating or copying of your colleagues code will result in severe penalties for all parties involved.
- Please try not a make any assumptions if you are unclear about any of the specifications please speak to me, either in class, office hours or via email, so that I can clarify any issues you may be having.