

Dictionary program

This program works as an online dictionary. It reads a word and shows its translation. The program loads the dictionary data from the “dict.dat” file.

The objective of the lab is to review the programming techniques learned in cs102.

The sections in parenthesis suggest where in “C++ for You++” to review.

- Building programs from source code (**compile and link**)(1.5)
- Communicating with the user using the console (**cin** and **cout**)(2.10)
- Creating and using **const** for data that will not change as program is running(3.6)
- Using boolean values(6.3)
- Reading data from a file into an array (or vector)(7.8)
- Using the 3 control structures: **sequence, selection, and iteration**(9.1)
- Searching an array (or vector) for a value(9.3)
- Using **pass-by-reference** in order to have a function “return” more than one value(11.3)
- Using **std::string** and **std::vector**(Appendix B, and 21.1)
 - The textbook shows us how to create these ourselves, but will use the built-in versions at first
- Create documentation using L^AT_EX(not in textbook)

Chapter 2 may be enough to get you started, but as soon and as much as you can, you should read all of *Part One : chapters 1-13*.

The focus of this semester is *Part Two : chapters 14-28*.

Dictionary program

This program works as an online dictionary. It reads a word and shows its translation. The program loads the dictionary data from the “dict.dat” file.

```
debian@debian:~/cs124/labs/dict$ ./lab
ENGLISH-ITALIAN DICTIONARY
(7 words)

Enter a word or 'q' to quit ==> love
amare

Enter a word or 'q' to quit ==> program
programma

Enter a word or 'q' to quit ==> q_
```

lab.h

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
struct ENTRY {
    std::string word;
    std::string translation;
};

// Function prototypes:
bool loadDictionary(std::string fileName, std::vector<ENTRY> &dict);
bool foundWord(const std::vector<ENTRY> &dict,
               const std::string &word, std::string &translation);
```

lab.cpp

```
#include "lab.h"
int main()
{
    std::vector<ENTRY> dict;
    std::string word, translation;
    bool ok, quit;

    ok = loadDictionary("dict.dat", dict);
    if (!ok) {
        cout << "*** Cannot load dictionary ***\n";
        return 1;
    }
    quit = false;
    while (!quit) {
        cout << "Enter a word or 'q' to quit ==> ";
        cin >> word;           // Read one word and
        cin.ignore(80, '\n');   // skip the rest of the line
        if (word == "q")
            quit = true;
        else if (foundWord(dict, word, translation))
            cout << translation << "\n\n";
        else
            cout << word << " -- not in the dictionary.\n\n";
    }
    return 0;
}
```

Load the dictionary from the file

Translate words

loadDictionary.cpp

```
#include "lab.h"
bool loadDictionary(std::string fileName, std::vector<ENTRY> &dict)
{
    int cnt = 0;
    std::string line;

    ifstream inFile(fileName.c_str());
    if (!inFile) return false;

    getline(inFile, line);
    cout << line << endl; // read header and ignore it
    ENTRY e;
    while (inFile >> e.word >> e.translation) {
        dict.push_back(e);
        inFile.ignore(80, '\n'); // Skip the rest of the line
        cnt++;
    }

    cout << " (" << cnt << " words)\n\n";
    cout << " (" << dict.size() << " size of vector)\n\n";

    return true;
}
```

Reads dictionary entries from a file.
Returns true if successful, false if cannot open the file.

Open dictionary file

Read and display the header line

Read words and translations into the dictionary array

Report the number of entries

foundWord.cpp

```
#include "lab.h"
bool foundWord(const std::vector<ENTRY> &dict,
               const std::string &word, std::string &translation)
```

- Finds a word in the dictionary.
 - dict – the dictionary array
 - word – word to translate
 - translation – returned translation from the dictionary.
- Returns true if the word has been found, false otherwise.

```
{
    bool found = false;
    int i, len = dict.size();

    for (i = 0; !found && i < len; i++) {
        if (dict[i].word == word) {
            translation = dict[i].translation;
            found = true;
        }
    }
    return found;
}
```

Makefile

```
src = lab.cpp loadDictionary.cpp foundWord.cpp
```

```
hdr = lab.h
```

```
CPPFLAGS = -g
```

```
exe: ${src:.cpp=.o} ${hdr}
```

```
    g++ ${src:.cpp=.o} -o lab
```

```
doc: ${src} ${hdr}
```

```
    latex lab.tex > log && tail log
```

```
    dvips -q lab.dvi
```

```
    ps2pdf lab.ps
```