CSCI 200 - Fall 2023
Foundational Programming Concepts & Design

Lab 3A - The Secret Moosage

🏠 **Home**   ✓☰ **HW Sets**   📅 **Schedule**   📥 **Files**   ⑦ **Help ▾**   🔗 **Links ▾**

This lab is due by Tuesday, October 10, 2023, 11:59 PM.
As with all labs you may, and are encouraged, to pair program a solution to this lab. If you choose to pair program a solution, be sure that you individually understand how to generate the correct solution.

Jump To: **Rubric Submission**

Concepts

The focus of this assignment is on how to read data from an "input file stream" or ifstream object and write data to an "output file stream" or ofstream object.

Working with Data

Reading From Files

Today's class discussed how data is often treated as "streams" of information that can be read a piece at a time. The files we will read in CSCI 200 are simple text files; for lab today, the simple text file contains a series of characters. Remember that whenever you work with a file stream as input, we call them `ifstream` objects.

There will always be four things you will do whenever working with an `ifstream`. Open the file, check for an error, read its data, and close the file. The typical pattern for this is as follows:

RMP-H

```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    ifstream myCatsAgesIn("filename"); // declare ifstream object and open the file

    // check for an error
    if ( myCatsAgesIn.fail() ) {
        cerr << "Error opening input file";
        return -1;
    }

    // read the data and do something with it
    int age;
    while( !myCatsAgesIn.eof() ) {
        myCatsAgesIn >> age;
        cout << "One cat is " << age << " years old.\n";
    }

    myCatsAgesIn.close(); // close the file
    return 0;
}
```

Remember, once you have an `ifstream` object (like `myCatsAgesIn` shown above) you use it in a manner similar to using `cin` .

Writing To Files

There will always be four things you will do whenever working with an `ofstream` .

1. Open the file
2. Check for an error
3. Write its data
4. Close the file

The typical pattern for this is as follows:

```cpp
#include <fstream>
#include <iostream>
using namespace std;

int main() {
    // declare ofstream object and open the file
    ofstream myCatsAgesOut("filename");

    // check for an error
```

```
        if ( myCatsAgesOut.fail() ) {
            cerr << "Error opening output file";
            return -1;
        }

        // write the data
        myCatsAgesOut << 5 << endl;
        myCatsAgesOut << 8 << endl;

        // close the file
        myCatsAgesOut.close();

        return 0;
    }
```

Remember, once you have an `ofstream` object (like `myCatsAgesOut` shown above) you use it in a manner similar to using `cout` .

Instructions

The cows have been kidnapped by aliens! The only clue to their whereabouts is a strange "ciphered" message, stored in the file **secretMessage.txt**. Fortunately, our in-house **cryptanalysis** expert, D. Cipher, has discovered the key:

"The key isn't very advanced. To decipher the message," she says, " you should take each character and replace all ~ (tilde) characters with a space, and shift all other characters up by one."

Your goal for this assignment is to create a program that reads the ciphered text file and then writes a deciphered version to a file called `decipheredMessage.txt` . For each character in the file, your program should implement the following replacement algorithm:

- If the character read is a newline character `'\n'` , then you should write the newline character to the file.
- Otherwise, if the character read is a ~ , you should write a space to the file.
- Otherwise, write the character read "offset" by +1.

To see if your implementation works, you should be able to open the file `decipheredMessage.txt` and see the information about the missing cows.

Hints

Where to place the input file

Any input file needs to be placed at the project level, which should be the same directory as your `main.cpp` file. You should see both your input file AND main.cpp file in the same directory.

Reading whitespace characters

In order to capture and replace whitespace characters, you will not use the `>>` operator with the input filestream. Instead, you will use the `get()` function like this:

```
while ( secretMessage.get(c) ) {
    // c is now assigned the next character from the file
}
```

This example assumes your `ifstream` is called `secretMessage` and you have a `char` variable declared that is called `c`.

Selection statement

Note that one requirement is to model the logic of the deciphering algorithm using a proper `if/else-if/else` construct. A `switch` statement could also be used.

Casting to `char`

Remember, `cout` and `ofstream` objects are sensitive to the datatype of the value to be printed or written to a file. Consider the following:

```
cout << ('a' + 2);
```

What is printed to the screen? The *number* 99. Why? Because a `char` plus an `int` yields an `int`, and then the int is "sent" to `cout`. To print the character `c` to the screen, you will need to use casting, like this:

```
cout << (char)('a' + 2);
```

Ahhh, much better.

Grading Rubric

Your submission will be graded according to the following rubric:

Points	Requirement Description
0.70	Fully meets specifications
0.15	Submitted correctly by Tuesday, October 10, 2023, 11:59 PM
0.15	**Best Practices** and **Style Guide** followed
1.00	**Total Points**

Lab Submission

Always, **always**, **ALWAYS** update the header comments at the top of your `main.cpp` file. And if you ever get stuck, remember that there is LOTS of **help** available.

Zip together your `main.cpp, Makefile` files and name the zip file `L3A.zip` . Upload this zip file to Canvas under L3A.

This lab is due by Tuesday, October 10, 2023, 11:59 PM.
As with all labs you may, and are encouraged, to pair program a solution to this lab. If you choose to pair program a solution, be sure that you individually understand how to generate the correct solution.

Last Updated: 09/08/22 12:39

Any questions, comments, corrections, or request for use please contact jpaone {at} mines {dot} edu.

[Jump to Top] [Site Map]