

# Laboratory V-B

CS-102

Spring 2022

# Saving Record Data in Text Files

- The program that follows will display a file called myfile.txt containing 6 records with two fields each.
- The data is as follows.
  - 420 "violet"
  - 470 "blue"
  - 500 "cyan"
  - 530 "green"
  - 580 "yellow"
  - 620 "orange"
  - 700 "red"
- The numbers refer to the wavelength, measured in nanometers, of each of the colors shown to the right.

```
// DisplayLightWavelength.cpp
#include <iostream>
#include <string>
#include <iomanip>
#include <fstream>
using namespace std;
int main()
{
    //reading from a file
    //create file objects
    ifstream infile;
    //need two variables to receive the data
    int lambda;
    string color;
    //associate them with a file
    infile.open("myfile.txt");
    //make sure the file exists
```

Program:  
DisplayLightWavelength.cpp  
Which is contained in the  
file: myfile.txt

```
if (!infile.is_open())           //file not found
    cout << "File not found" << endl;
else
{
    //read first value
    Infile >> lambda;
    while (!infile.eof())
    {
        //read the rest of the line
        infile >> color;
        //print what we just read
        cout << "The wavelength of " << color << " is "
            << lambda << " nanometers" << endl;
        //read the beginning of the next data record
        infile >> lambda;
    }
    infile.close();
}
return 0;
}
```

# DisplayLightWavelength .cpp

When the program is run, the output is:

The wavelength of violet is 420 nanometers

The wavelength of blue is 470 nanometers

The wavelength of cyan is 500 nanometers

The wavelength of green is 530 nanometers

The wavelength of yellow is 580 nanometers

The wavelength of orange is 620 nanometers

The wavelength of red is 700 nanometers

# Laboratory V-B part 1

- Write a program called: *YourName-LabVB-1.cpp* which takes the contents of *myfile.txt*, and creates a new file called: *results.txt*
- *results.txt* is a textfile which has the format:  

color frequency
- color simply refers to the color of the record being processed.
- frequency refers to the frequency (given in TeraHertz [THz]) of the color.
- Remember that  $\text{frequency} = \text{SpeedOfLight} / \text{WaveLength}$ .
- Note that although the WaveLength was given as an integer, the frequency will need to be given as a double.
- Use the following value for the speed of light: 299,792.46 km/sec.

# Laboratory V-B part 1

- When you have *YourName*-LabVB-1.cpp working:
  - If you are doing LabVB synchronously, please demonstrate your program and results.txt to the instructor for proper credit.
  - If you are doing LabVB asynchronously, please submit your program and results.txt to Canvas.

# Laboratory V-B part 2

- Write a program called: ***YourName-LabVB-2.cpp*** , which takes the contents of results.txt, and displays those results so that you can clearly see what the frequency of each of the colors of the spectrum happens to be.
- For each of the colors of the spectrum the output should be of the following format:

The frequency of *color* is *xxx.xx THz*

- When you have *YourName-LabVB-2.cpp* working:
  - If you are doing LabVB synchronously, please demonstrate your program to the instructor for proper credit.
  - If you are doing LabVB asynchronously, please submit your program to Canvas.



# Laboratory V-B Part 3

- Expanding on the concepts introduced in parts 1 and 2, prompt the user for the name of an input file and the name of an output file.
  - Make sure to verify that the input file exists,
  - Use a loop to validate and get another name.
  - You do not have to verify that the output file exists because it will be created.
- BEFORE you start to process the data, MAKE SURE THAT YOU CAN JUST READ AND PRINT ALL THE DATA IN THE FILE!
- Remember your program should work for a file with any number of records without modifying your code.
- Create one file of your own with at least 5 entries that you create.
  - Show the results with the sample file given, and your own file.
- For each problem, show your source code, the input file, and the output file to the instructor to receive credit.

# Laboratory V-B Part 3

- **Problem 3:** Write a program to process employee pay increases.
- The data will be in a file with the following format:
- Employee\_name current\_salary percent\_increase.txt
- Here is the content of **datafile.txt** which you will make (note 1 space only twixt entries):
  - Miller,Andrew 63700 5.5
  - Green,Shelia 55025.75 6.2
  - Abrams,Milton 57400 6.1
  - Smith,Stacey 72900 5.2
  - Charles,Ray 54095 6.3
- Note that there must be NO SPACES in the employee name. As a result, you will be able to read using >> .
- Your program will ask for:
  - The name of the file to process, and the name of the output file.
  - Read the employee name, current salary, and increase.
  - Calculate the new pay, and save the result in a file called **salaryIncrease.txt**.

# Laboratory V-B part 3

- When you have your file **datafile.txt** created and ***YourName-LabVB-3.cpp*** working:
  - If you are doing LabVB synchronously, please demonstrate your program to the instructor, and by showing the resulting file, **salaryIncrease.txt**, for proper credit.
  - If you are doing LabVB asynchronously, please submit your files **datafile.txt** and **salaryIncrease.txt** as well as your program, ***YourName-LabVB-3.cpp***, to Canvas.

# Laboratory VB Part 4

- Now take your file, **salaryIncrease.txt**, and write a program which will display the contents of the file to the screen.
- Call your new program: ***YourName-LabVB-4.cpp***
- Format for the display of the input file (here are the first two outputs):

Salary Increase Report:

<u>Employee</u>	<u>New Salary</u>
Miller,Andrew	\$67203.50
Green,Shelia	\$58437.35

- If you are doing LabVB synchronously, please demonstrate your program to the instructor, and show the resulting output from ***YourName-LabVB-4.cpp***
- If you are doing LabVB asynchronously, please submit your file: **salaryIncrease.txt** as well as your program, ***YourName-LabVB-4.cpp***, to Canvas.

# Laboratory VB Part 5

- You will now create a new file, **salaryIncrease2.txt**, and write a program which will display the contents of the file to the screen.
- Call your new program: ***YourName-LabVB-5.cpp***
- Below is the new data for **salaryIncrease2.txt**, and how we want ***YourName-LabVB-5.cpp*** to display it.

<u>Employee</u>	<u>New Salary</u>
Miller,Andrew	\$67203.50
Green,Shelia Gay	\$58437.30
Abrams,Milton	\$60901.40
Smith,Stacey Anne	\$76690.80
Charles,Ray	\$100000.00

- What you will immediately notice, from the new data, is that now there are names that have spaces in them. This means that you will need to handle them differently because the stream extraction operator, `>>`, that we have been using does not work with spaces, but terminates operation when one occurs.
- What we will do is use the `#` sign (it could be any non-alphanumeric character) to separate the string part of the data in **salaryIncrease2.txt** from the numeric part instead of a space as we did in **salaryIncrease.txt**.
- Given this new data, now write the file: **salaryIncrease2.txt** .

# Laboratory VB Part 5 cont.

- You will now need to use **getline(infile, name, '#');** instead of **infile >> name;** where '#' is the delimiter that separates the string part of the output from the numeric part of the output. (Actually, most any non-alpha character may be used as a delimiter.)
- However, you will still be able to use **infile >> newSalary;** as before.
- But, this introduces a new problem, which we've seen before when we mix **getline()** outputs with numeric outputs which use: **>> .**
  - The spacing will be altered due to the difference between the stream extraction operator, **>>**, and the **getline()** function. Try it and see what I mean.
  - The cure, as usual, is to follow every instruction where we use **infile >> new Salary;** immediately with the instruction **infile.ignore();**
- If you are doing LabVB synchronously, please demonstrate your program to the instructor, and show the resulting output from **YourName-LabVB-5.cpp**
- If you are doing LabVB asynchronously, please submit your file: **salaryIncrease2.txt** as well as your program, **YourName-LabVB-5.cpp**, to Canvas.

# Laboratory VB Part 6 – Extra Credit

- You will now create a new file, **salaryIncrease2.txt**, and write a program which will display the contents of the file to the screen.
- Call your new program: ***YourName-LabVB-6.cpp***
- Below is the new data for **salaryIncrease2.txt**, and how we want ***YourName-LabVB-5.cpp*** to display it.

<u>Employee</u>	<u>New Salary</u>
Miller,Andrew	\$67203.50
Green,Shelia Gay	\$58437.30
Abrams,Milton	\$60901.40
Smith,Stacey Anne	\$76690.80
Charles,Ray	\$100000.00

- What you will immediately notice, from the new display, is that we want the decimal point to line up, in the output, as well as the \$ being next to the most significant digit. To do this, we will need to use the `<sstream>` library.
- Rewrite the display program you used in part 5 to use stringstream. You will likely want to consult your notes from today's lecture on how to handle stringstream within a loop.
- Use the file: **salaryIncrease2.txt** , that you created for part 5.