Thomas Gibbons

October 26, 2016

ECE 3220

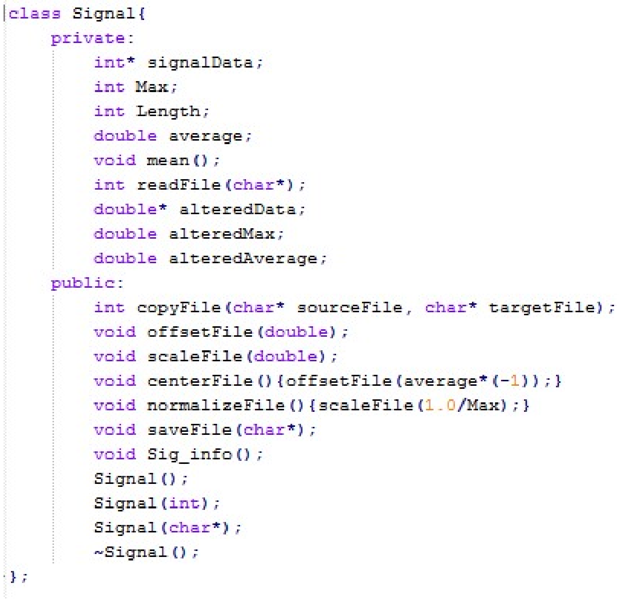
LAB 7

Objective

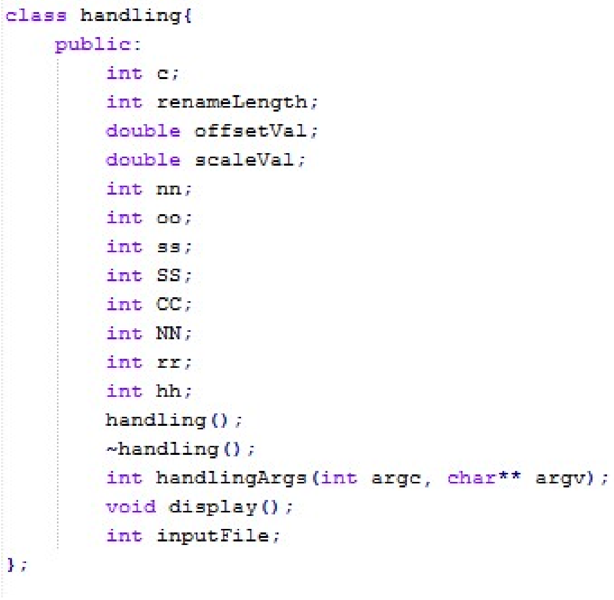
The objective of this lab was to become familiar with C++ using classes and objects. We learned to use default constructors, parametric constructors, and destructors. We learned how to overload functions putting different inputs. We also furthered our knowledge of exception handling.

Discussion

In this lab we converted lab 4 and 5 C codes into C++ codes. I created two classes. One was a signal class that was required and stored the sample data, max, length and average as well as an array for the altered data regardless of how it was being altered. It also contains all the functions from the previous lab and turned them into methods of the class. The next was a handling class where I made everything public, but was there just to keep track of all the variables and I made a few methods in it. The constructor for it set all the variables. During testing I used the display method so I could see the handling was working and flags were being set appropriately. The big method was the handlingArgs which basically took all of the error checking from the beginning of the previous lab and put it in one method. That method set all the variables in the class appropriately.

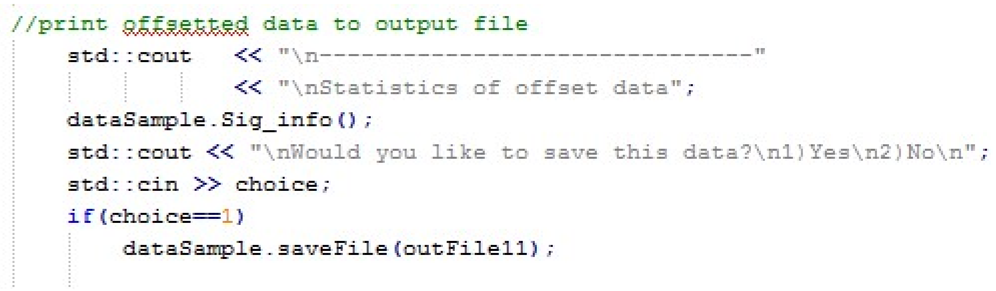


```
class Signal{
    private:
        int* signalData;
        int Max;
        int Length;
        double average;
        void mean();
        int readFile(char*);
        double* alteredData;
        double alteredMax;
        double alteredAverage;
    public:
        int copyFile(char* sourceFile, char* targetFile);
        void offsetFile(double);
        void scaleFile(double);
        void centerFile(){offsetFile(average*(-1));}
        void normalizeFile(){scaleFile(1.0/Max);}
        void saveFile(char*);
        void Sig_info();
        Signal();
        Signal(int);
        Signal(char*);
        ~Signal();
};
```



```
class handling{
    public:
        int c;
        int renameLength;
        double offsetVal;
        double scaleVal;
        int nn;
        int oo;
        int ss;
        int SS;
        int CC;
        int NN;
        int rr;
        int hh;
        handling();
        ~handling();
        int handlingArgs(int argc, char** argv);
        void display();
        int inputFile;
};
```

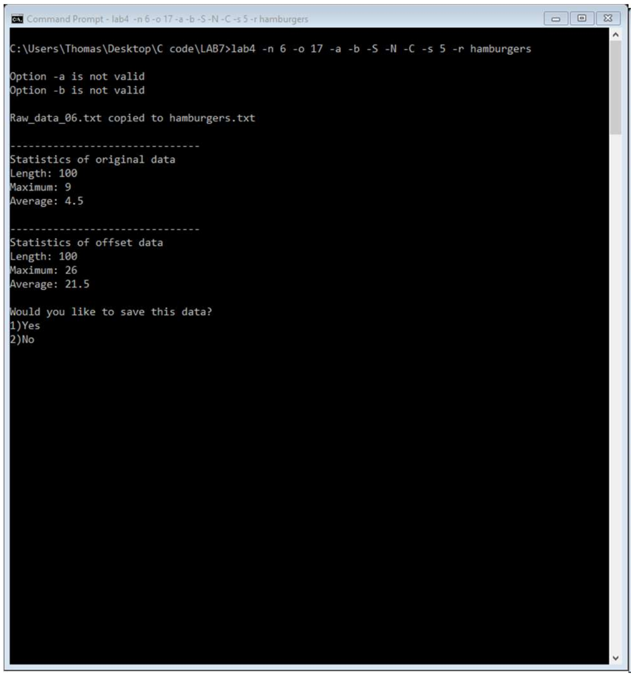
The logic behind the main is very similar to that of the previous lab. The biggest difference is that whenever outputFile was called in the previous lab it was switched with displaying the current details of the altered data using the Sig_info method and then giving the user a choice if they want the data saved as shown below.



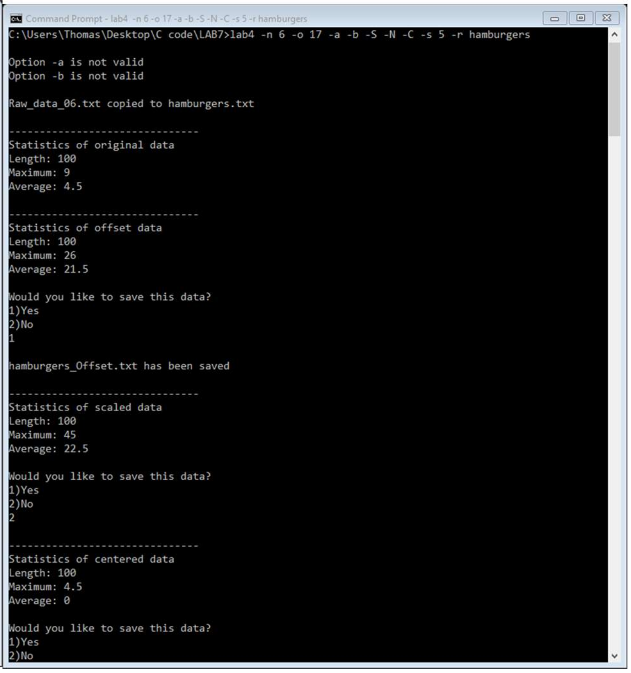
```
//print offsetted data to output file
    std::cout   << "\n------------------------------"
                << "\nStatistics of offset data";
    dataSample.Sig_info();
    std::cout << "\nWould you like to save this data?\n1)Yes\n2)No\n";
    std::cin >> choice;
    if(choice==1)
        dataSample.saveFile(outFile11);
```

Another big difference is, as shown above, we did not use printf or scanf but instead used the C++ equivalent in using cout and cin. This in itself was a big part of the lab going through tediously changing them. Also changing malloc to new and free to delete[] was a big change.

Here are some output screenshots. This is an example of using every handling option. It shows the invalid option. It displays the current altered signal info and then asks if you want to save it. If yes it will name it appropriately and show that it was saved, else it will move to the next option requested.

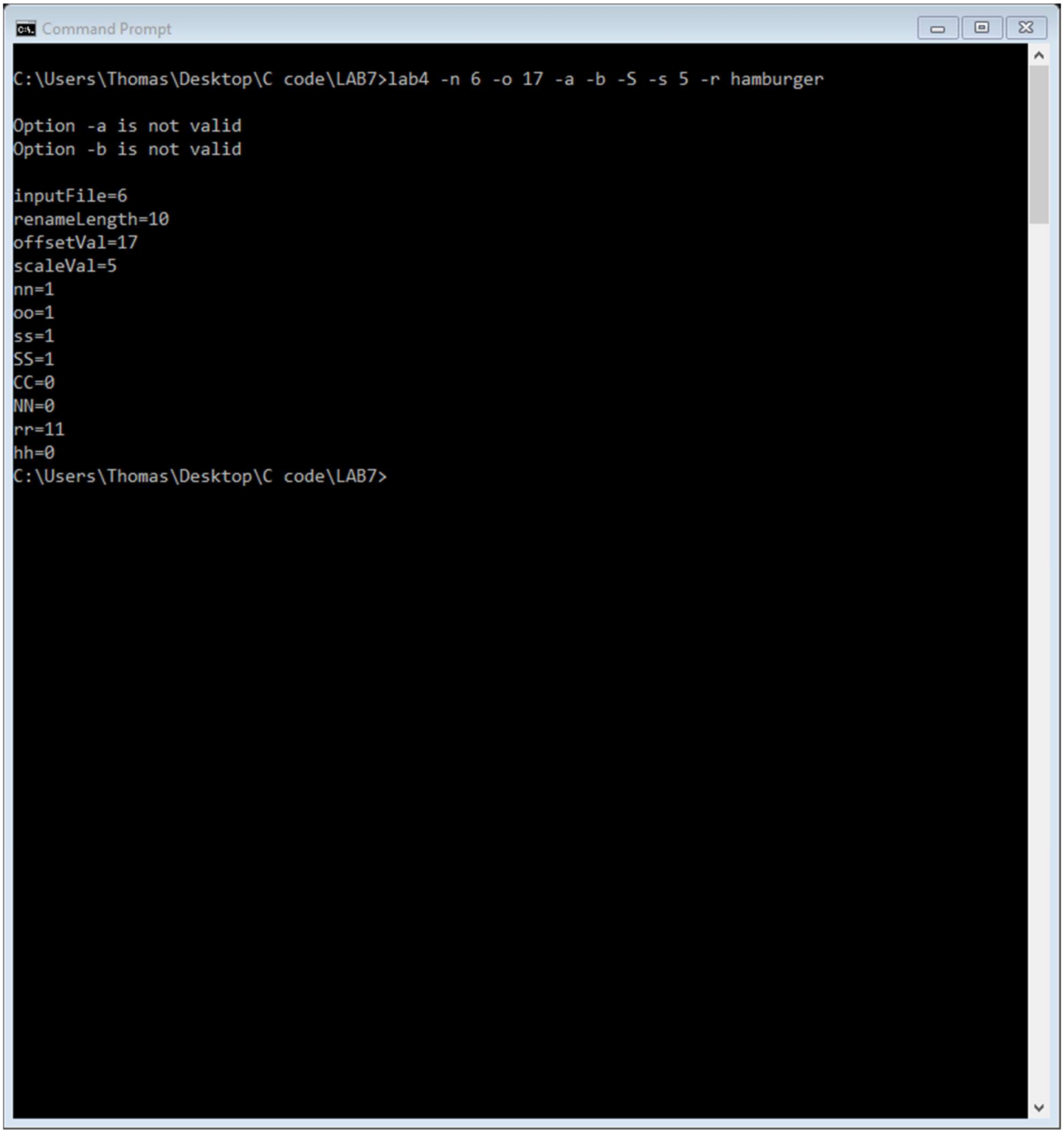

Command Prompt - lab4 -n 6 -o 17 -a -b -S -N -C -s 5 -r hamburgers

```
C:\Users\Thomas\Desktop\C code\LAB7>lab4 -n 6 -o 17 -a -b -S -N -C -s 5 -r hamburgers

Option -a is not valid
Option -b is not valid

Raw_data_06.txt copied to hamburgers.txt

------------------------------
Statistics of original data
Length: 100
Maximum: 9
Average: 4.5

------------------------------
Statistics of offset data
Length: 100
Maximum: 26
Average: 21.5

Would you like to save this data?
1)Yes
2)No
```


Command Prompt - lab4 -n 6 -o 17 -a -b -S -N -C -s 5 -r hamburgers

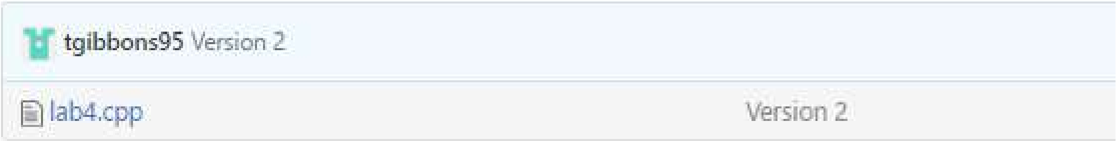
```
C:\Users\Thomas\Desktop\C code\LAB7>lab4 -n 6 -o 17 -a -b -S -N -C -s 5 -r hamburgers

Option -a is not valid
Option -b is not valid

Raw_data_06.txt copied to hamburgers.txt

------------------------------
Statistics of original data
Length: 100
Maximum: 9
Average: 4.5

------------------------------
Statistics of offset data
Length: 100
Maximum: 26
Average: 21.5

Would you like to save this data?
1)Yes
2)No
1

hamburgers_Offset.txt has been saved

------------------------------
Statistics of scaled data
Length: 100
Maximum: 45
Average: 22.5

Would you like to save this data?
1)Yes
2)No
2

------------------------------
Statistics of centered data
Length: 100
Maximum: 4.5
Average: 0

Would you like to save this data?
1)Yes
2)No
```

Results of showing the flags using the display in the handling is shown below. This shows the arguments that are being used is file number, offset, scale, Statistics, and rename. Each is given a 1 to signify their use except rename is given the index of where the new name is so in this case argv*[11] = “hamburger”. Shows the length of the new name, the input file number, offset value, scale value, and invalid arguments.



```
C:\Users\Thomas\Desktop\C code\LAB7>lab4 -n 6 -o 17 -a -b -S -s 5 -r hamburger

Option -a is not valid
Option -b is not valid

inputFile=6
renameLength=10
offsetVal=17
scaleVal=5
nn=1
oo=1
ss=1
SS=1
CC=0
NN=0
rr=11
hh=0
C:\Users\Thomas\Desktop\C code\LAB7>
```

As always a link to my github account is https://github.com/tgibbons95/Lab6.



tgibbons95 Version 2

lab4.cpp Version 2