

Overview

The purpose of the project is to give you additional C++ practice. For this assignment, you may work individually or in a group with one other student. You will write a program which reads a text file containing 32-bit (type `uint32_t`) unsigned hexadecimal integers. Your program will create a histogram of the numbers in the file. A histogram is a count of the number of times each value occurs. Your program will also count the number of values in the file, the number of unique values, and the number of lines, and will find the smallest number on each line.

Setup

In CodeLite on your virtual machine:

- Create a new C++ project named Project2.
- Add a class named HexHistogram (files HexHistogram.h and HexHistogram.cpp) to this project. To do this in CodeLite, right click on the src folder and select New Class. Fill in the name of the class and click "OK." (All default setting in the New Class dialog are fine.)
- Notice that CodeLite automatically creates both the .cpp and .h files for your class. In addition, it generates a default constructor and a destructor in the .cpp file and a `#ifndef` in the header file.

Submitting Your Assignment

Compress (zip) the src folder of your CodeLite project. You will need to do this using your operating system's file manager because CodeLite does not provide a way to zip the source folder. (It's probably easier to do this on your host computer. This will work as long as your workspace is in the shared folder.) Name the zip file YourLastNameYourFirstNameProject2.zip. Upload this zip file to Canvas before the due date.

Important submission note for students working in a group:

If you worked with another student, name the zip file:

FirstPersonsLastNameSecondPersonsLastNameProject 2.zip

For example, if I was working with Daisy Duck, I would name my zip file SherbaDuckProject2.zip. One person should upload the zip file to Canvas. The other person should upload a single document stating who their team member was.

Remember that late assignments are not accepted in this course. See the syllabus for details.

Assignment

Write a C++ program with the following:

1. A class named HexHistogram (files HexHistogram.h and HexHistogram.cpp)
 - Function prototypes and member variables must be declared in the header (.h) file.
 - All member variables must be private.

- The constructor and all member functions must be implemented in HexHistogram.cpp.
- The constructor has one parameter of type `std::string`, the name of a text file.
 - Your class must read the file only once, in the constructor. The constructor must store the needed information (see below) in appropriate variables and data structures. The constructor must close the file.
 - Any errors encountered in the constructor (such as input file not found) should throw an exception of type `std::runtime`. A string describing the error must be supplied as the exception argument. The string must also include the file name.

Your class must implement these public methods:

- `get_value_count`: returns the total number of values in the file (type `uint32_t`).
- `get_unique_value_count`: returns the number of unique values in the file (type `uint32_t`).
- `get_line_count`: returns the number of lines in the file (type `uint32_t`).
- `get_smallest_number(uint32_t line_number)` returns the smallest value in the line (as an `uint32_t`)
 - The first line in the file is line 1.
 - If an invalid line number is given, you must throw a `std::logic_error`.
- `print`: print the histogram contents
 - Each unique value must be printed followed by a colon (:), followed by the number of occurrences of the value.
 - The values must be printed in increasing order of the unique values. Values must be printed as hexadecimal numbers. The values must be right aligned in a field of width 8.
 - The number of occurrences must be right aligned in a field of width 8. The number of occurrences should be printed as a decimal number.

2. A main function in main.cpp

The main function accepts a single command line argument of type `std::string`, the name of a text file.

- If the number of command line argument is incorrect, you must print the following message:

```
Usage:  ./Project2 <filename>
```

Your main program must create a single instance of HexHistogram using the input file specified as the command line argument. The main program must then print the results of functions `get_value_count`, `get_unique_value_count`, and `get_line_count`.

All counts must be output as decimal numbers. All numbers read from the file must be output as hexadecimal numbers. Your output should be formatted so that it is easy to read.

The program must then use a loop to print the line number followed by a colon (:), followed by the smallest value in that line. You must start with the last line in the file and end with the first

line in the file. Line numbers must be printed as decimal numbers and must be right aligned in a field of width 4. The values must be right aligned in a field of width 8 and printed as hexadecimal numbers.

After printing the statistics above, call the print method of the HexHistogram class.

Hint: To see all of your output, you may need to change the number of scrollbar lines in your terminal window. You can find this setting by clicking on the “Edit” menu, choosing “Preferences,” and then clicking on the “Display” tab.

Implementation Notes for WordCount

Use the C++ type `ifstream` (http://en.cppreference.com/w/cpp/io/basic_ifstream) for the input file.

Use the function `getline` (http://en.cppreference.com/w/cpp/string/basic_string/getline) to read each line of the file into a string.

For each line, create a variable of type `istringstream` (http://en.cppreference.com/w/cpp/io/basic_istringstream) to hold the line contents for parsing numbers.

Parse the numbers by using the `>>` operator with the `istringstream` variable and a string destination. Use the `std::hex` modifier to read the values as hexadecimal numbers.

Both the `getline` function and the `>>` operator return a value which can be tested as true or false in an *if* or *while* statement. A false value indicates end of file or error.

You should use the `std::map` class (<https://en.cppreference.com/w/cpp/container/map>) to store the frequency count (histogram) data and the smallest number data.

Sample Input/Output

Input:

```
334 92f    011a 92f
ffff0000    1ffff
1234abcd dedc9876 11a
dedc9876 92f    01ffff
```

Output:

```
Number of values read: 12
Number of unique values read: 7
Number of lines: 4
```

Smallest number on each line

```
4:      92f
3:      11a
2:     1ffff
```

1: 11a

Histogram

11a:	2
334:	1
92f:	3
1ffff:	2
1234abcd:	1
dedc9876:	2
ffff0000:	1

Grading Criteria (20 points possible)

Note: Your program must compile to receive credit for this assignment!

Points	Criteria
0-1 points	Structure: Does the program contain the three files described above? Are function prototypes and member variables declared in the header (.h) file? Are all member variables private? Are all member functions implemented in HexHistogram.cpp?
0-2 points	Correctness: Is the constructor implemented as described in the assignment? Does it have one parameter of type std::string? Is an exception thrown if the file cannot be opened? Does it read the file only once?
0-1 point	Correctness Is the get_value_count member function implemented as described in the assignment? Does the function return the correct value?
0-1 point	Correctness Is the get_unique_value_count member function implemented as described in the assignment? Does the function return the correct value?
0-1 point	Correctness Is the get_line_count member function implemented as described in the assignment? Does the function return the correct value?
0-3 points	Correctness Is the get_smallest_number member function implemented as described in the assignment? Does the constructor store the correct value for each line in a map? Does the function return the correct value? Is a std::logic_error exception thrown if an invalid line number is passed to the function?
0-2 points	Correctness: Is the print member function implemented as described in the assignment? Does the constructor store the correct values in a map?
0-2 points	Output: Is the histogram printed as described in the assignment? Are values printed in increasing order? Are values printed as hexadecimal numbers and number of occurrences printed as decimal numbers? Is all data right aligned in fields of width 8?
0-1 point	Correctness: Is the main function implemented as described in the assignment? Does it accept one command line argument of type std::string? Is a "usage" error message printed if the number of command lines arguments is not correct?
0-1 point	Output: Does the main function output all requested counts as decimal numbers?
0-2 points	Output: Does the main function use a loop to output the line number and smallest number on each line as described in the assignment? Are the lines printed in reverse order, starting with the last line and ending with the first? Are line

	numbers printed as decimal numbers and right aligned in a field of width 4? Are all values output as hexadecimal numbers and right aligned in a field of width 8?
0-1 point	Output: Does the main method call the print method to output the histogram?
0-2 points	Style: Is the code easy to read? Does it follow class style guidelines (See Program Style page in Canvas.)