Programming Assignment 1 Documentation

CS 3377.0W1

Basma Mahamid

**Documentation of Cpp Code:**

I separated my code into functions:

The *findToken()* function which is a generic function I used on multiple occasions throughout the program. It is a general function that takes a string, two characters, and a current Index (as a reference variable) as parameters, and returns a string. The string that is passed as the parameter is the full URL, and the current index parameter tracks and increments and updates the index that the function stops at as it is continuously used throughout the scope of the main function and is incremented. The character parameters are to pass the character (or in the case of the optional port) that the function should stop at this character and separate and return this substring in order to find the components of the URL. I also used *findToken()* in my *validateDomain()* function in order to separate the domain into three substrings separated by periods.

I also have functions for each part of the URL to check that they are valid. They are all boolean functions that pass that specific part of the URL as the parameter and return true if that specific part is valid. I used the *.compare()* function to compare strings to compare the substring to what is allowed/valid. The *validateDomain()* function not only ensures that the third part of the domain is valid but also that there are 3 total components. The *validatePort()* function checks to ensure that the port is within the allowed range.

At the end of the main function, I wrote an if statement that says if any of the validation functions return false meaning there is an error, a general error message arises and lists only the parts of the URL that were invalid. If there were no errors, each component of the URL is listed next to what component of the URL it is, the Protocol (Scheme), Domain, Port, File path, parameters.

**Documentation of Linux Code:**

After I started the script recording with the command *script* followed by what I named the script file, I began writing the commands in the terminal as follows:

I was in the subdirectory I created, /Homework/Assignment \_1, so I used the command *ll* which lists the names of all the files in this current directory along with the subdirectories.

I already had imported my coded .cpp file into the subdirectory so after doing this I used the command *g++* followed by the *.cpp file name* which compiles and creates an executable file with the default name a.out but I wrote the command *-o* which changes the name of the executable file from this default name to whatever I followed that command with the name *“URLParser.cs2”* to change the name of the executable file.

I used the command *ll* again to once again see the files in the directory to show my executable I just created at the bottom of the list.

Now in order to execute/ run and test that file I used the command *./* followed by the name of the executable to run.

Then the program runs and I used multiple different user inputs to test the program-

I used 3 valid URLS (in accordance with the program guidelines) as well as 3 invalid URLS. Each time the program runs, in order to run it again I use the same command “./URLParser.cs2” to run the program again.