CSCE 221 Assignment 4 Cover Page

First Name: Joseph Last Name Martinsen UIN 323009961

User Name josephmart E-mail address josephmart@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: `http://aggiehonor.tamu.edu/`

Type of sources	Stack Overflow			
People				
Web pages (provide URL)				
Printed material				
Other Sources				

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work. *On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*

Your Name Joseph Martinse Date 03/30/2017

CSCE 221 — Programming Assignment 4, Spring 2017 (100 points)

Due: March 30th at 11:59 pm

Report

Joseph Martinsen
PA4
File Parsing and Regex

File Parsing and Regex – Assignment Description (100 points)

A very common task in Computer Science is the reading in and parsing of text. One of the most powerful tools at a programmer's disposal to aid in this task is regex (which stands for REGular EXpressions).

1. What is stored in "`matches`"?
   99
2. What does "`\d`" mean?
   Digit
3. Modify the regex pattern to retrieve a two-digit number and the word `thanks` in the string. Test your pattern for correctness.
   Check the code
4. (25 points) Compile and run the following code, then answer the questions:
   (a) What does "`\s\S`" mean?
   "`\s`" means white space
   "`\S`" means no white space.
   (b) What is stored in `matches[0]`?
   `matches[0]` is "<title>This is a title</title>"
   (c) Why is `matches[1]` different?
   Because there are two regexs patterns in "R"(<title>([\s\S]+)</title>)"
   (d) Modify the regex pattern to retrieve only the items inside of the header tag but not inside of the title tag. Test your pattern for correctness.
   Check the code

5. (40 points) Download the following text file: stroustrup.txt

   Write a program using regex that will go through the text file and print out the file name of every hyperlinked powerpoint file. (Hint1: an HTML hyperlink uses the format `<a href=”...>...</a>`. Hint2. The powerpoint file extension is `.ppt`)
   Check the code

- Your C++ source code with the header block including: your name, user name, section number and e-mail address

```
// CSCE 222 - 506
// Name:     Joseph Martinsen
// Username: josephmart
// Email:    josephmart@tamu.edu

#include <iostream>
#include <string>
#include <regex>
#include <fstream>
using namespace std;

int main() {
    // Part 3
    std::cout << "Part 3" << '\n';
    regex pattern5{R"(\d\d)"};
    regex pattern6{R"(thanks)"};
    string to_search0 = "I would like the number 98"
              " to be found and printed, thanks.";
    smatch matches5;
    smatch matches6;
    regex_search(to_search0, matches5, pattern5);
    for (auto match : matches5) {
        cout << match << endl;
    }

    regex_search(to_search0, matches6, pattern6);
    for (auto match : matches6) {
      cout << match << endl;
    }

    std::cout << "\n\n";
    // Part 4
    std::cout << "Part 4" << '\n';
    regex pattern{R"(<head>([\s\S]+)<title>)"};
    regex pattern1{R"(</title>([\s\S]+)</head>)"};
    string to_search = "<html><head>Wow such a header <title>This is a title</title>"
                    "So top</head>Much body</html>";
    smatch matches0;
    smatch matches1;
    regex_search(to_search, matches0, pattern);
    cout << matches0[1] << endl;
    regex_search(to_search, matches1, pattern1);
    cout << matches1[1] << endl;


    std::cout << "\n\n";

    // Part 5
    std::cout << "Part 5" << '\n';
    std::ifstream afile("../stroustrup.txt", std::ios::in);
    std::string allData;

    if (afile.is_open()) {
       std::string line;
       while (std::getline(afile, line)) {
            allData += line;
            allData += ’\n’;
       }
       afile.close();
   }
   else {
       std::cerr << "Unable to open file\n";
   }

   regex pattern2{R"(<a href=\"(.*?.ppt)\")"};
   smatch matches2;

   // Shoutout to StackOverflow
   std::string::const_iterator searchStart(allData.cbegin());
   while(regex_search(searchStart, allData.cend(), matches2, pattern2))
    {
        std::cout << ( searchStart == allData.cbegin() ? "" : "\n" ) << matches2[1];
        searchStart += matches2.position() + matches2.length();
    }
    std::cout << ’\n’ << ’\n’;
}
```

- Description of input and output data. List all restrictions and assumptions that you have imposed on your input data and program.

  Input data is an html page and output is powerpoint names.

- Write your regex patterns used for parsing the strings in the programs above. Explain their syntax.

  Part 3: Search for digit then digit for the first part. Next it looked for string literal `thanks`

  Part 4: Had two regex patterns. First it searched for any white-space or non-white-space from title to head. Next regex pattern searched for any white-space or non-white-space from head to title.

  Part 5: This part finds characters between <a href=" and " that end in ppt. This search loops, once a match has been found, it will then continue searching.

- What is the purpose of the functions `regex_search()` and `regex_match()`.

  `regex_match()`
  Checks whether there is a match between the regular expression e, and all of the character sequence

  `regex_search()`
  Checks if there is a sub-sequence in the string that matches the regular expression.

- Which C++ features or standard library classes have you used in your program?

  ```
  #include <iostream>
  #include <string>
  #include <regex>
  #include <fstream>
  ```