**Student ID:** 985848\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Full Names:** Van Quy Tran\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Web Application Programming

(CS472)

(September 2017)

Instructor: Obinna A. Kalu

W1D6 – Exam 1

1. The exam duration is 2 hours.
2. The exam is a computer-based exam.
3. You are expected to use a CS lab or your own computer to answer both the Coding questions and the theory/non-coding/knowledge-based questions. You may use the Internet and/or the lecture slides for reference purposes to lookup APIs or code syntax.
4. Make sure to switch-off your cell-phones or simply turn the ringer off.
5. You may use blank sheet(s) of paper for your scratch work, if needed.
6. Exams are copyrighted materials and must not be copied, reproduced or distributed.
7. All answers to the theory/non-coding/knowledge-based questions should be typed-in, on this document, following the questions.
8. All answers to the Coding questions may be typed-in as source code files, using a Code Editor or IDE. But be sure to copy your finished code for each coding question from your Code Editor and paste it to this document as your answer.
9. Finally, compress/zip your entire code folder into one zip file and upload/submit it to Sakai, along with your typed/pasted answers in this document.

--------------------------------------------------------------------------------------------------------------------

(CS472 - WAP)

(September 2017)

W1D6 – Examination 1

**Part I – Science of Consciousness (SCI):** (3 points)

From Maharishi’s teachings, we learnt about the principles of the Science of Consciousness which are the fundamental laws of nature that uphold all progress and growth in life. In the last 1 week, you have learnt about several concepts and principles in the Web Application Programming (WAP) course. In the table below, three (3) Web programming topics are given, on the first column. For each of these, write-down in the 2nd column, a related principle from the Science of Consciousness which you are familiar with, and in the 3rd column, write 1 or 2 sentences to show how the Web Programming topic relates to your chosen SCI principle.

**Note**: To get the full credit, make sure your sentence(s) provide(s) a reasonable, clear connection between your stated SCI principle and the given WAP topic. You may use a relevant analogy from Maharishi’s teachings, to help illustrate your point.

|  |  |  |
| --- | --- | --- |
| **Topic from Web Application Programming** | **SCI principle** | **Connection sentence(s)** |
| The Internet, the World-Wide Web and HTML | The benefit of TM is that it settles the mind so that it more easily becomes aware of the deeper layers affecting the relative. | We have several layers in TCP/IP like Application, Host to Host, Internet, Network Access  Same as the layers of consciousness |
| Regular Expressions (Regex) | Purification leads to progress. | In every application we need to verify the data that user inputted in frontend side and backend side also. So we can make sure the data is correct and the system is secure. |
| Cascading Style Sheets (CSS) | Seek the highest first. | The frontend is the face of the website and we need a lot of CSS to make our website look good. In CSS we need to layout from the most outer element to inner element like the principle in SCI “Capture the Fort” |

**Part II – Theory (Short answers, True/False, Multiple-choice questions):** (37 points)

1. (7 points) Answer the following questions with True or False. For each answer, give a rationale (i.e. If True state how, if False state why. No rationale, earns you just half of the points if your True/False answer is correct, and zero point if your True/False answer is incorrect).
   1. (3 points) In order to create a sub-form inside a bigger Web-form, an HTML <form> element can be contained or nested inside another HTML <form> on a web page.

False.

We can have multiple forms on a single webpage, but forms CANNOT be nested.

* 1. (2 points) The HTML <table> element can be used to create complex Web page layouts.

True.

Using tables to build structure is quite intuitive. We see tabular data every day, and the concept is well known. We can use nested table for complex layout. But nowadays, we shouldn’t use table anymore because Tables generally increase the complexity of documents and make them more difficult to maintain. Also, they reduce a website’s flexibility in accommodating different media and design elements, and they limit a website’s functionality. Instead, we can use div with float, position, flexbox or grid.

* 1. (2 points) Based on the HTML markup and CSS styling presented in the figure P2 – 1 – III below, the <div> with class named, mainmenu, will be **positioned** on the webpage at 200pixels from the left margin and 100pixels from the top margin.



Figure P2 – 1 - III

False.

Because *static* is the default position value for all elements.

An element with position: *static*; is not positioned in any special way.

A *static* element is said to be not positioned and an element with its position set to anything else is said to be positioned.

We can use position: *relative* or *absolute* to achieve the margin above.

1. (23 points) Give short answers to the following questions.
   1. (4 points) What is the difference between the 2 CSS selectors given below?

**A.**

div > p {

…

}

**B.**

div p {

…

}

**Answer 2 - I:**

A: Only target the *p* that are direct descendants of *div*

B: Target all *p* are descendants *div*.

* 1. (4 points) What is the difference between the 2 CSS properties settings,

**visibility: hidden** versus **display: none**?

**visibility: hidden** will hide the element, but the element will still take up the space it would if it was fully visible

**display: none** will render the page as though the element does not exist.

* 1. (3 points) What is the original purpose/function of the CSS *float* property?

The reason that the *float* property became so common was that, by default, block-level elements will not line up beside one another in a column-based format. Since columns are necessary in virtually every CSS layout, this property started to get used.

The float property specifies whether or not a box (an element) should float.

The CSS float property allows a developer to incorporate table-like columns in an HTML layout without the use of tables.

* 1. (4 points) What do we mean by a **Semantically correct HTML markup** (You may give an example to illustrate your answer).

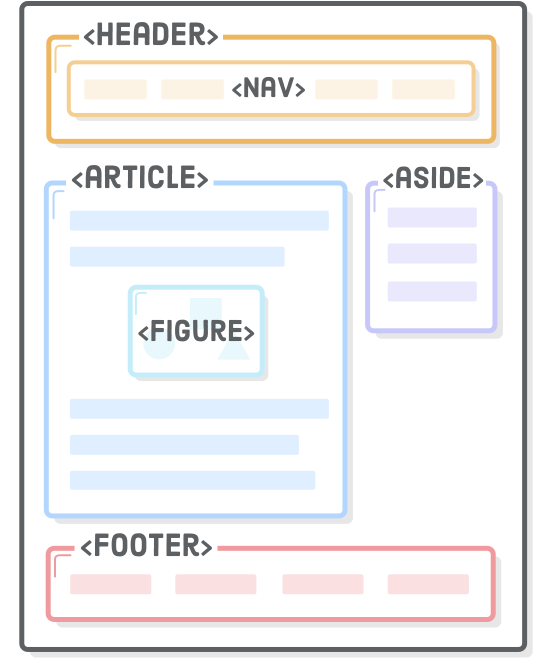
A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.

Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

Using these as an alternative to <div> elements is an important aspect of modern web development because it makes it easier for search engines, screen readers, and other machines to identify the different parts of your website. It also helps you as a developer keep your site organized, which, in turn, makes it easier to maintain.

Example:



* 1. (4 points) Write a relative URL that would take you from <http://mumstudents.org/cs472/~98009999/test/index.html> to: <http://mumstudents.org/cs472/register.html>.

<a href="../../register.html"Register</a>

* 1. (4 points) Write a regular expression (regex) that matches any string that starts with the word, HTML followed by at least one numeric character.

HTML\d+

1. (7 points) The following questions involve multiple choices; choose the correct option by putting a green highlight over, either Option A or Option B or Option C.
   1. (2 points) For an HTTP Request received by a web server and processed without any error, the HTTP Response code sent back is:

**Option A**.

404

**Option B**.

200

**Option C**.

500

* 1. (3 points) Which is the incorrect HTML markup for implementing Radio buttons on a Web Form:

**Option A**.

<label><input type=**"radio"** name=**"prefcar"** value=**"Toyota"** checked=**"checked"**/>

**Toyota**</label>

<label><input type=**"radio"** name=**" prefcar"** value=**"Chevrolet"** /> **Chevrolet**</label>

**Option B**.

<label><input type=**"radio"** name=**"toyota"** value=**"Toyota"**checked=**"checked"**/>

**Toyota**</label>

<label><input type=**"radio"** name=**"chevrolet"** value=**"Chevrolet"** /> **Chevrolet**</label>

* 1. (2 points) Which is the incorrect regular expression for matching US Social Security numbers formatted as XXX-XX-XXXX (where X is number)?

**Option A**.

^\d{3}-\d{2}-\d{4}$

**Option B**.

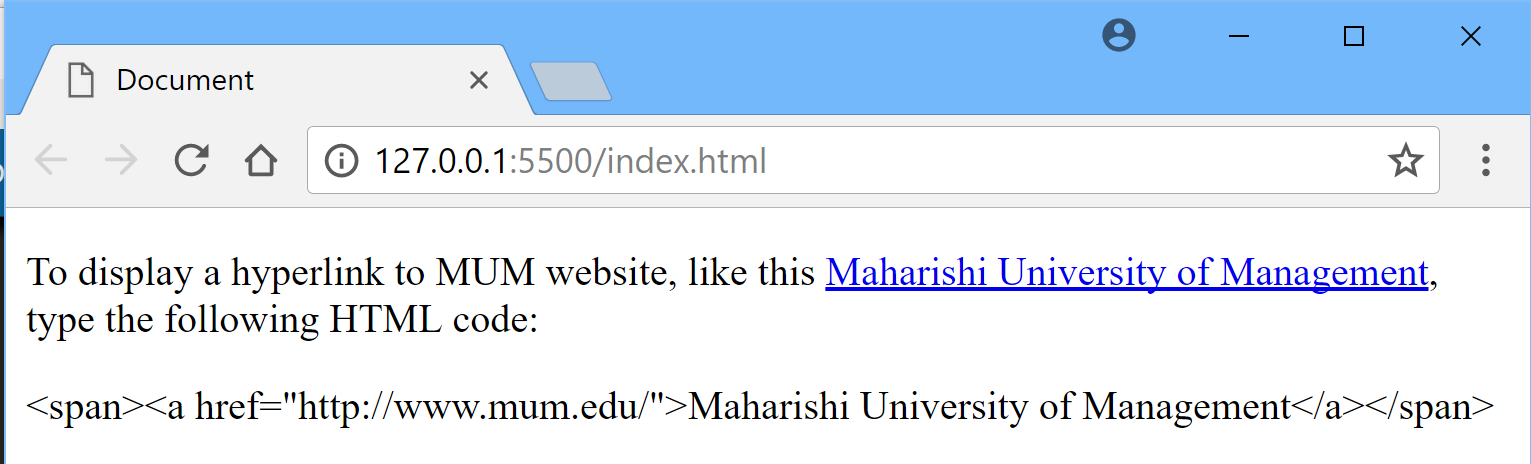
\d{3}-\d{2}-\d{4}

**Option C**.

\d{3}\-\d{2}\-\d{4}

**Part III – Skill (Web Coding):** (60 points)

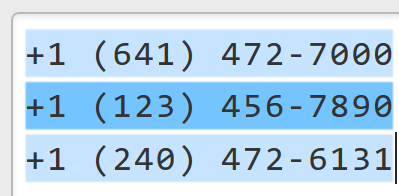
1. (10 points) Create a web page (filename: index.html) that displays the content as shown in the browser screenshot given below.



**Answer 1:**

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
</head>  
<body>  
 <p>To display a hyperlink to MUM website, like this <span><a href="http://www.mum.edu/">Maharishi University of Management</a></span>  
 , typing the following HTML code:</p>  
 <p>&lt;span&gt;&lt;a href=&quot;http://www.mum.edu/&quot;&gt;Maharishi University of Management&lt;/a&gt;&lt;/span&gt;</p>  
</body>  
</html>

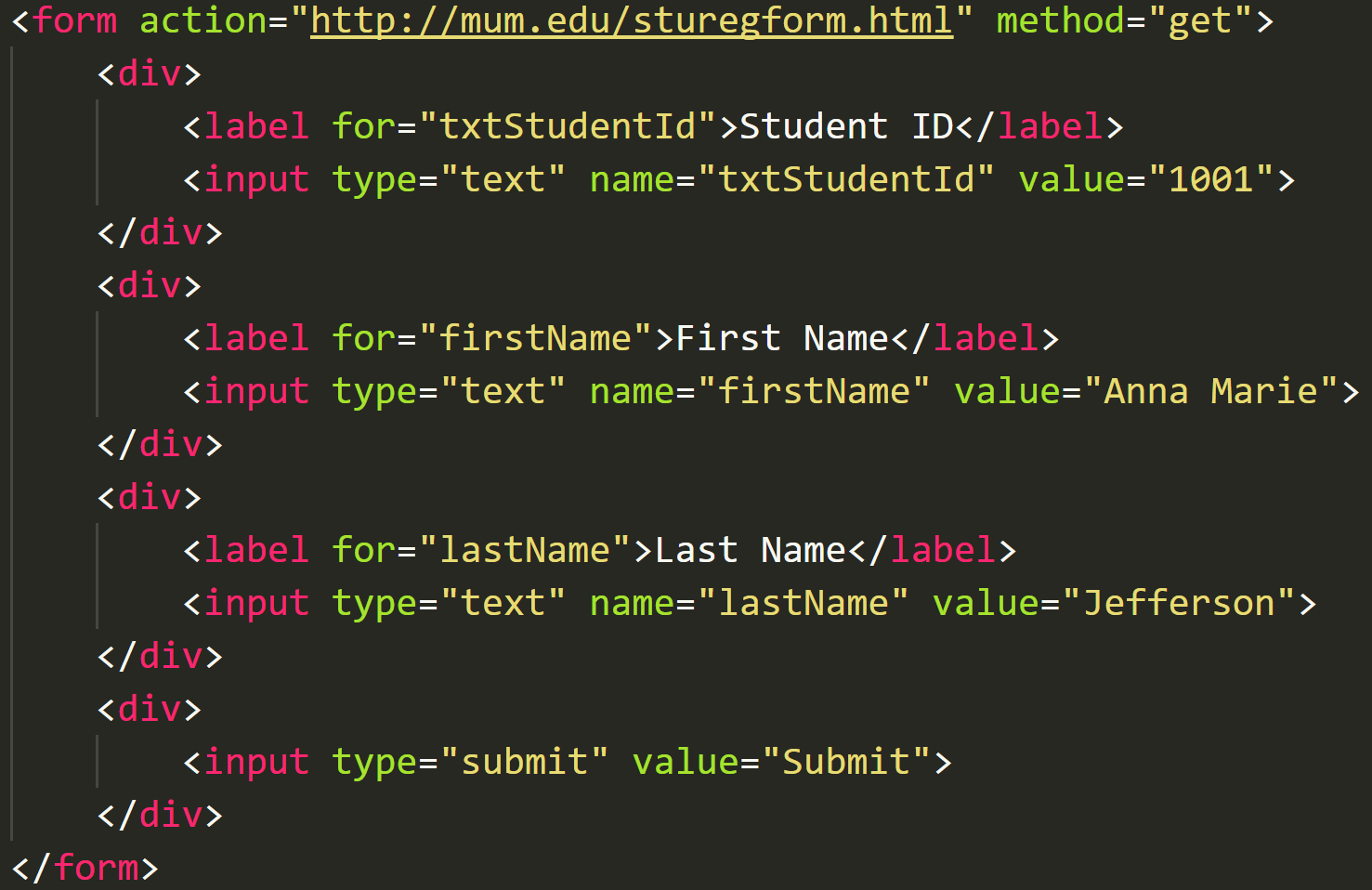
1. (10 points) Answer the following questions:
   1. (5 points) Write an appropriate Regular Expression to match valid United States telephone numbers, formatted as shown in the figure below, including the country code, +1.



**Answer 2a:**

\+1\s\(\d{3}\)\s\d{3}-\d{4}

* 1. (5 points) Consider the HTML markup for a Web Form shown below.

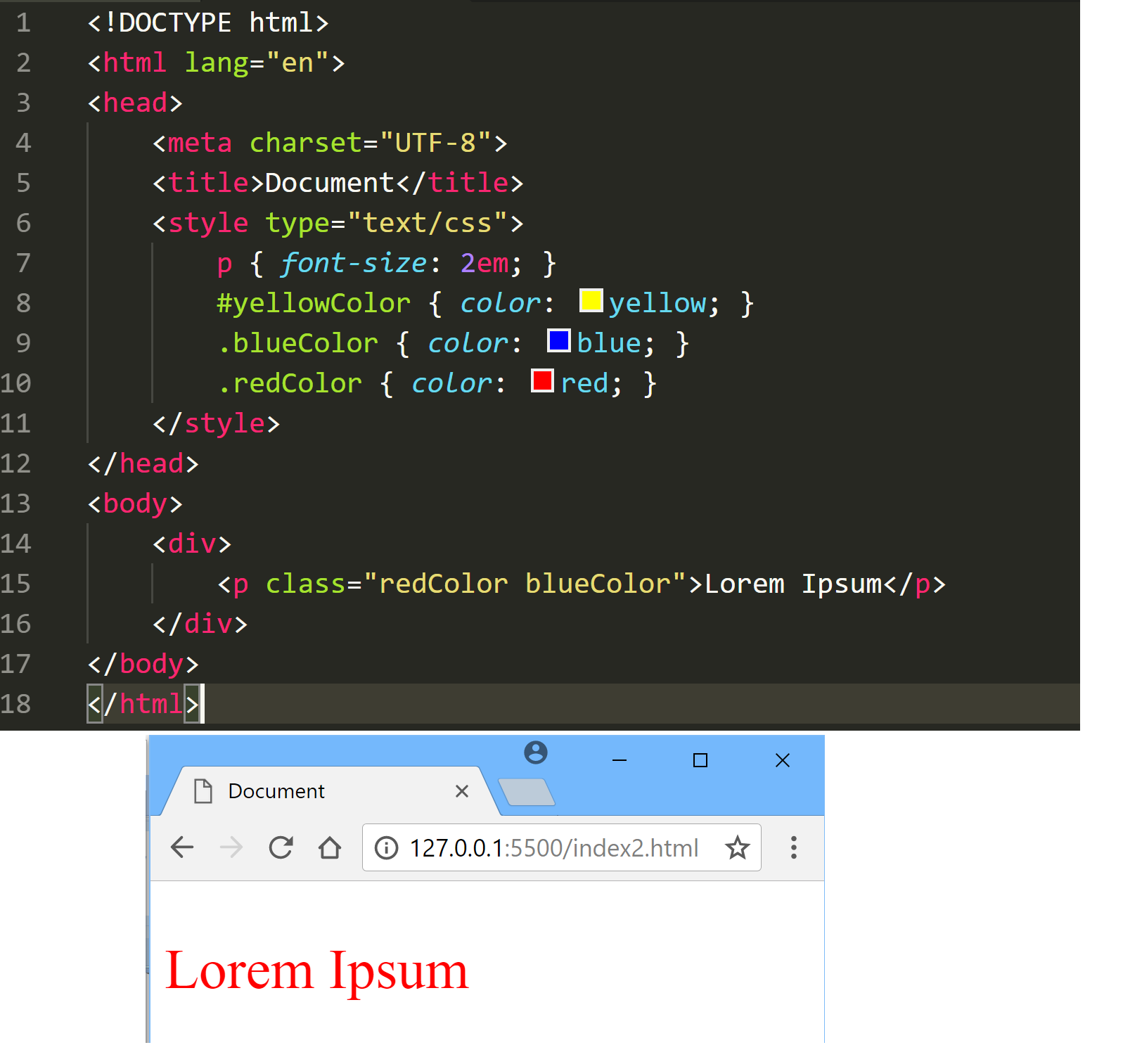


Write-out the full destination URL (including the query string/parameters) generated by the browser, when the form is submitted to the web server.

**Answer 2b:**

https://www.mum.edu/sturegform.html?txtStudentId=1001&firstName=Anna+Marie&lastName=Jefferson

1. (10 points) Answer the following questions:
   1. Consider the HTML markup displayed below, along with the browser output (also displayed below), which is a paragraph content with the text, “Lorem Ipsum”, colored, red.



**Task 1** - (5 points): Re-create the HTML markup, and include/make appropriate modification to the embedded CSS styling such that the paragraph content with the text, “Lorem Ipsum”, is displayed in blue.

**Answer 3a – Task 1:**

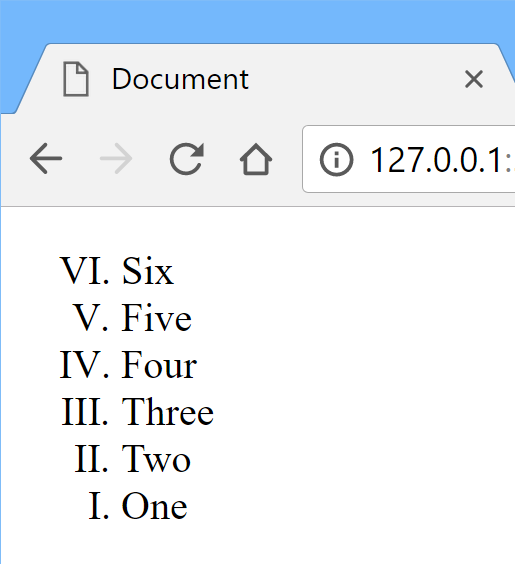
<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <style type="text/css">  
 p { font-size: 2em; }  
 #yellowColor { color: yellow; }  
 .redColor { color: red; }  
 .blueColor { color: blue; }  
 </style>  
</head>  
<body>  
 <div>  
 <p class="redColor blueColor">Lorem Ipsum</p>  
 </div>  
</body>  
</html>

**Task 2** - (5 points): Re-create the HTML markup again, and include/make another (different) appropriate modification to the embedded CSS styling such that the paragraph content with the text, “Lorem Ipsum”, is displayed in blue.

**Answer 3a – Task 2:**

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <style type="text/css">  
 p { font-size: 2em; }  
 #yellowColor { color: yellow; }  
 .blueColor { color: blue **!important**; }  
 .redColor { color: red; }  
 </style>  
</head>  
<body>  
 <div>  
 <p class="redColor blueColor">Lorem Ipsum</p>  
 </div>  
</body>  
</html>

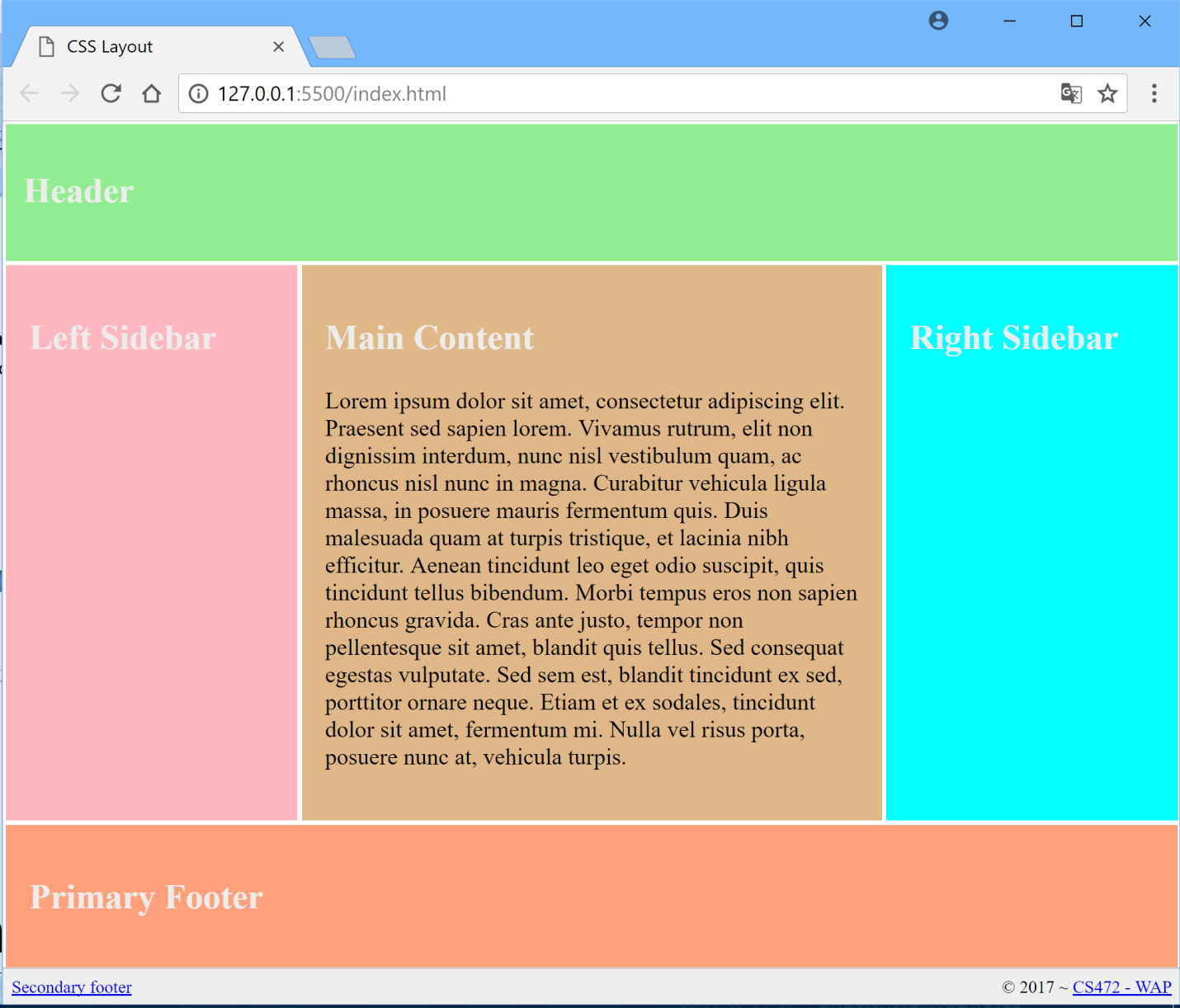
1. (10 points) Write HTML markup and appropriate CSS styling, to display a web page with a list of Numbers displayed in reverse order and using roman numerals, as shown in the figure below:



**Answer 4:**

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <style type="text/css">  
 ol {  
 list-style-type: upper-roman;  
 transform: rotate(180deg);  
 }  
  
 ol > li {  
 transform: rotate(-180deg) translateX(40px);  
 }  
 </style>  
</head>  
<body>  
 <ol>  
 <li>One</li>  
 <li>Two</li>  
 <li>Three</li>  
 <li>Four</li>  
 <li>Five</li>  
 <li>Six</li>  
 </ol>  
</body>  
</html>

1. (20 points) Using an appropriate Web layout technique, write HTML markup and CSS styling to produce the page layout, as shown in the figure below. *(Please note: The aim is to produce the given page layout structure. So, feel free to use any colors and text content of your choosing*):



<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <style type="text/css">  
 \* {  
  
 box-sizing: border-box;  
 }  
 .container {  
 display: flex;  
 flex-direction: column;  
 color: #ffffff;  
 }  
  
 .flex-item {  
 margin: 2px;  
 padding: 10px;  
 }  
  
 .container header {  
 background-color: #157ffb;  
 }  
  
 .container .section-1 {  
 display: flex;  
 flex-direction: row;  
 overflow: hidden;  
 }  
  
 .container .section-1 .left-sidebar {  
 background-color: sienna;  
 flex: 0 0 25%;  
 }  
  
 .container .section-1 article {  
 background-color: deeppink;  
 flex: 0 0 50%;  
 }  
  
 .container .section-1 .right-sidebar {  
 background-color: tomato;  
 flex: 0 0 25%;  
 }  
  
 .container .footer-1{  
 background-color: springgreen;  
 }  
  
 .container .footer-2{  
 background-color: palevioletred;  
 display: flex;  
 flex-direction: row;  
 }  
  
 .container .footer-2 span {  
 flex: 0 0 50%;  
 }  
  
 .container .footer-2 span:last-child {  
 text-align: right;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <header class="flex-item">  
 <h1>Header</h1>  
 </header>  
 <section class="section-1">  
 <aside class="left-sidebar flex-item">  
 <h1>Left Sidebar</h1>  
 </aside>  
 <article class="flex-item">  
 <h1>Main Content</h1>  
 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas vitae volutpat ante, ut aliquam nibh. Quisque tincidunt magna sapien, bibendum mattis odio ullamcorper eget. Nullam est nunc, vulputate at aliquet consequat, imperdiet nec ante. Morbi sit amet elit eget quam auctor condimentum quis quis dui. Integer in ex non quam eleifend bibendum nec ut ex. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec a enim nec velit semper malesuada vel eget leo. Duis ut ipsum ut mi dignissim accumsan et id leo. Maecenas faucibus sed erat nec pharetra. Nulla pretium lorem quis egestas finibus. Aliquam in lacus vitae nulla placerat vestibulum. Nulla sed orci faucibus, vulputate purus in, auctor ligula. Donec quis vestibulum orci. Cras id velit sit amet orci rhoncus condimentum eget sit amet ipsum. Duis tempus sed enim at iaculis. Donec augue leo, egestas eget odio eget, feugiat tincidunt sem.</p>  
 </article>  
 <aside class="right-sidebar flex-item">  
 <h1>Right Sidebar</h1>  
 </aside>  
 </section>  
 <footer class="footer-1 flex-item"><h1>Primary Footer</h1></footer>  
 <footer class="footer-2 flex-item">  
 <span><a href="#">Secondary footer</a></span>  
 <span>&copy; 2017 - <a href="#">CS472 - WAP</a></span>  
 </footer>  
 </div>  
</body>  
</html>

**//-- The End --//**