COS 301 Programming Languages Fall 2012
Project Assignment #3 Due Tuesday Nov 6
Data Types, Expression Evaluation and Assignment Statements

Write 7-to-10 page paper that addresses the following topics for your selected language:

- Data types available **in the language** beyond primitive data types such as ints, floats, etc. This might include arrays, vectors, structures, strings, pointers, objects and classes. You should include higher-level structures such as hashes, stacks, etc. **if they are part of the language definition rather than part of a standard or other library.**
- Summary of standard libraries or classes that extend the type system.
- Semantics of expression evaluation
- Coercions or implicit type conversions that are performed in expression evaluation
- Semantics of assignment statements

Programming Assignment
HTML Keyword Markup

Write a function that accepts an HTML document (a string) and a keyword (also a string). The function finds all occurrences of the keyword string in the HTML after the <body> element (provided that the string does not occur within an html tag) and surrounds the string with tags that will "highlight" the keyword; for example:

```
<span style="background-color: blue; color: white">keyword</span>
```

Most browsers highlight text in an HTML document when you use the Find command (often Ctrl-F)
You do have to careful not to mark up strings that are found inside an HTML tag. For example, if the search string is "table" you do not want to mark up this:

```
<table width="100%" border="0">
```

because this is not valid HTML:

```
<<span style='background-color: blue;'>table</span>width="100%" border="0">
```

Two approaches to this problem:

1. String processing. Treat the HTML as one long string and program a lexer that returns the next chunk; that chunk being either a tag, text (data between tags), or a comment (<!-- this is a comment -->). If it is a tag or a comment, simply append it to output. If it is text, search the text for the string and apply the markup if found.

The string approach is not the best way to approach this problem; among other things due to the complexity of HTML and XHTML. To be correct for all html documents, the content of certain elements should be skipped, such as <script>, <object>, <embed>, <textarea>. For the purposes of this assignment this level of refinement is not necessary. The string approach also treats one abstraction (HTML document) as another type of abstraction (string).

If you use the string processing approach, don't search within the <head>. Start your search after "<body".

2. DOM Document processing.

An HTML document is handled a browser and other document processors by transforming the text into a tree structure defined by the W3C Document Object Model (DOM). For out purposes, each node in a DOM document is either an HTML element, a comment, script or text (we'll ignore other node types such as CDATA nodes). To handle string markup, traverse the DOM tree and check the text nodes. You can also use an XPath expression to do actual string search and return a set of nodes. Either way, if you find a string to mark up, remove the text node, and create and append a new subtree (left text, markup node with child text, and the right text).

Although this sounds difficult it is in fact quite easy to do with the DOM parser libraries that can be obtained for virtually any language –just search for [language name] DOM parser.

You are welcome to use either approach. If you are not familiar with HTML and the DOM then the string processing approach is undoubtedly easier in the short run. However the DOM approach is much easier programming, and if you make the effort to understand the DOM you will acquire the foundation for a set of skills that will be useful for years.

Write a program that demonstrates and tests your function. If your language is a web scripting language, you can simply use a form that posts to your program, which outputs the html back to the browser. Or you can write program that reads data from an HTML file and outputs the revised HTML to a different file.

For output submit the first page of the HTML source of the project policy and format at

http://umcs.maine.edu/~cmeadow/courses/cos301/project-policy.html

with the keyword "paper" highlighted. You can also try a search for "left" in this page to verify that your program does NOT mark up hits inside HTML. The page has four instances of

```
<p style="padding-left: 5%">
```