## Assignment-1

## Background

For your first assignment, you are to create your own home page that will be placed on the Web. In addition to telling the world about you, your home page will serve as a central repository of Web-based applications that you develop for this module.

## Submission of work

**NOTE:** As per course teacher.

Assessment

**This assignment contributes 1/3 of the 25% continuous assessment part of the lab taks. The maximum possible grade on this assignment is 100 points.** Failure in this task can be compensated by higher marks on the other assessments of the module.

## Learning outcomes

The purpose of this assignment is for you to demonstrate the following learning outcomes and for me to assess your achievement of them.

Students should:

1. have an understanding of the range of programming techniques and languages available to organisations and businesses and be able to choose an appropriate architecture for a web application.
2. be able to demonstrate abilities to design and implement maintainable web sites.
3. be able to make informed and critical decisions regarding client development using HTML and JavaScript.
4. have the knowledge to critically analyse and evaluate web applications.

## 

## 

## Description of the task

Aside from the required elements below, the layout and content of your Web page are entirely up to you. You are encouraged to be creative and produce a document that you are proud to display to the world and that is easy to read.

At a bare minimum, your page must provide the following:

### Part 1 (35 points)

* Your page should have at least two different parts, one containing some personal information (like your name, where you're from, etc.), and another set of links. You should include both internal links (that jump to locations within the info page) and external links that would lead to other pages on the web (like something related to one of your hobbies, etc).
* Your info page must have your name prominently centered at the top.
* There should be several sections to the info page (e.g., Personal Info, Hobbies, Favorite Links), each with its own heading. At least one section must have multiple paragraphs in it.

### 

### Part 2 (15 points)

* Your info page should perform some text formatting (bold, italics, color, BIG) but try not to overdo it to the point that the content is compromised.
* You should include a favorite quote, e.g., a poem or a line from a movie.
* There should be at least one list (e.g., your top-ten favorite CDs).
* There should be at least 5 links in the page, including a link to the University of Liverpool and to the COMP 519 home page. At least one of those links must occur in a sentence.

### Part 3 (25 points)

* There should be at least 2 images in the page, one of which must be stored locally along with the Web page.
* There must be at least one table, some of whose rows are more than one line long.
* The page's location and last modification date should be automatically displayed at the bottom. (Use JavaScript here.)
* Your page must make use of a style sheet that defines basic formatting style, including the format of tables. The style sheet should be defined in a separate file and linked to your page. You can also use some inline directives for a few of the simple styles (like making **bold-faced** text).

**Important note:** Your page should use standard HTML (and CSS) features only, so that it will be viewable using Internet Explorer, Netscape (Firefox), Opera, Safari, Chrome, etc. While it may not look exactly the same due to inconsistencies and variations in the browser implementations, your page should look reasonable under any such browser.

### Part 4 (25 points)

Your homepage should have a nice (professional) look and its code should satisfy common standards. In particular, for maximum points here, your web page should satisfy the Strict XHTML 1.0 (or 1.1) standards, or the HTML5 standards (including a proper Document Type Declaration in the HTML file). You should pass it through the [W3C validator](http://validator.w3.org/). If you have any errors, you might try to run your HTML code through [tidy](http://www.w3.org/People/Raggett/tidy/) (although this is no excuse for sloppy coding in the first place).

## 

## Assignment-2

## Background

The second assignment focuses on JavaScript, and the development of several features/wepages using JavaScript to demonstrate your understanding of that language and its usage in web development.

## Submission of work

**NOTE:** As per course teacher.

Assessment

**This assignment contributes 1/3 of the 25% continuous assessment part of the course grade. The maximum possible grade on this assignment is 100 points.** Failure in this task can be compensated by higher marks on the other assessments of the module.

## Learning outcomes

The purpose of this assignment is for you to demonstrate the following learning outcomes and for me to assess your achievement of them.

Students should:

1. have an understanding of the range of programming techniques and languages available to organisations and businesses and be able to choose an appropriate architecture for a web application.
2. be able to demonstrate abilities to design and implement maintainable web sites.
3. be able to make informed and critical desicisions regarding client development using HTML and JavaScript.
4. have the knowledge to critically analyse and evaluate web applications.

## Description of the task

**Note:** The specific look and feel of the pages described below is left intentionally vague, allowing considerable design freedom on your part. However, for full points the pages should have a nice look and its code should satisfy common standards (meaning, for example, that it would be easy for someone else to read your HTML, CSS, and JavaScript code and **understand** what the intention of the pages is, how the JavaScript code is supposed to operate, and **it should be relatively straightforward to maintain**).

**Important note**: It is assumed as part of this assignment that the HTML code you write will satisfy the HTML5 standards!! (Include an appropriate [Document Type Declaration](http://cgi.csc.liv.ac.uk/~martin/teaching/comp519/HANDOUTS/Document-types.pdf) and consider using the [W3C online validator](http://validator.w3.org/).) Failure to satisfy the HTML5 standards will result in losing marks from your grade.

### Part 1 (30 points)

Add a random "fortune generator" to your home page. That is, your page should contain a list of fortunes (stored as an array of strings), and should randomly select one of those fortunes to display each time the page is loaded. The fortune should be displayed just above the page footer, centered and enclosed in a box. Here's an example given below.

|  |
| --- |
| True wisdom comes not from knowledge, but from understanding. |

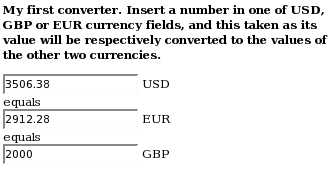
Please put your JavaScript in an external file, link to it in the <head> element of your page, and just put an appropriate function call on your homepage to display the fortune.

### Part 2 (35 points)

**You can assume current exchange rates amongst several currencies or find it in google.**

For example, 1 USD = 84.00BDT and 1 CAD = 1.01941 USD.

Create a Web page named convert.html that can be used to convert some value in one currency to the respective values of the other currencies. That is, your page should include a form which consists of five currency fields, and inserting a number in one of the fields, this number, taken as the respective value of the currency, be converted to the values of the other currencies. For example, a page of USD, EUR and GBP converter might look like this (the converted values don't match the values in the table above):



### Part 3 (35 points)

In order to get all 35 points for this part, create a Web page named quiz.html that can be used to conduct multiple choice quizzes over the Web. The page should contain at least 10 potential quiz questions, each with three possible answers (A, B, and C). When loaded, the page should first prompt the person for the number of desired questions in the quiz, with a default of 5 questions. The page should then randomly select questions and prompt the user with each question and possible answers. Each answer entered by the user should be compared with the correct answer, and the result displayed within the page (either CORRECT or INCORRECT). At the end, the number and percentage of correct guesses should be displayed in the page.

Your page must be clear and understandable to the user, and support the following:

* It must be straightforward to add or remove potential questions, with a minimal amount of editing.
* It must display each question and all three potential answers as part of the prompt, and make it clear to the user how their answer should be formatted.
* The question, user's answer, and correctness of that answer must be displayed in the page. In the case of an incorrect answer, the correct answer must be identified.
* The number of correct answers, total number of questions, and correctness percentage must be displayed in a readable format.
* For extra credit, ensure that no question appears more than once in a specific quiz. Of course, this requires enough potential questions to complete the quiz.

For example, the page might contain the following as a result of a 5-question quiz:

What is the capital of Missouri?

You guessed B) Jefferson City

CORRECT

How many ounces in a pound?

You guessed A) 10

INCORRECT: the correct answer is C) 16

Who was the first person to set foot on the moon?

You guessed C) Neil Armstrong

CORRECT

Who holds the Major League Baseball record for most home runs in a season?

You guessed A) Barry Bonds

CORRECT

In what year was University of Liverpool founded?

You guessed A) 1250

INCORRECT: the correct answer is B) 1881

You answered 3 out of 5 questions correctly (60%).

### 

### Remarks:

1. There's loads of information about JavaScript that's available online, more than I can possibly tell you about in the lectures.
2. The third part of this assignment is likely the most difficult (so do the other two parts first).
3. You are definitely able to do all these tasks using only JavaScript (and HTML, CSS of course).
4. The functionality of your pages (at least for the second and third parts) relies on writing appropriate event handlers to control the tasks that you want to perform.
5. You will likely want to remind yourself/read more about the JavaScript method called "[getElementById](http://www.w3schools.com/jsref/met_doc_getelementbyid.asp)" which can be very useful to retrieve various elements of a webpage (using the ids that you assign to them). This then allows you to extract the values from those elements (like input text boxes, check boxes, radio buttons, etc) and/or assign new values to them.
6. It's also possible to use the "[innerhtml](http://www.w3schools.com/jsref/prop_html_innerhtml.asp)" property to change values (but *this can sometimes not work correctly based on particular browers or versions*, so read more about it if you choose to use this way to alter page contents). An alternative method to alter webpage contents dynamically is to use the methods of creating and appending (or deleting) the nodes of the [Document Object Model](http://www.w3schools.com/js/js_htmldom.asp).
7. For the quiz part (the third part), there are several ways in which you might approach this. If you don't want to worry about the varying number of questions, feel free to just create a quiz that has ten questions (but this won't get you all of the 35 points).
8. Whatever method you choose, whether it's a straightforward method, or a more complicated approach (by manipulating the DOM), my suggestion is to proceed slowly and test your code a lot along the way.

## 

## Assignment-3

## Learning outcomes

The purpose of this assignment is for you to demonstrate the following learning outcomes and for me to assess your achievement of them.

Students should:

1. have an understanding of the range of programming techniques and languages available to organisations and businesses and be able to choose an appropriate architecture for a web application.
2. be able to demonstrate abilities to design and implement maintainable web sites.
3. be able to make informed and critical decisions regarding client development using HTML and JavaScript.
4. be able to design and implement reasonably sophisticated server-side applications using one or more suitable technologies.
5. have the knowledge to critically analyse and evaluate web applications.

## Submission of work

**NOTE:** As per course teacher.

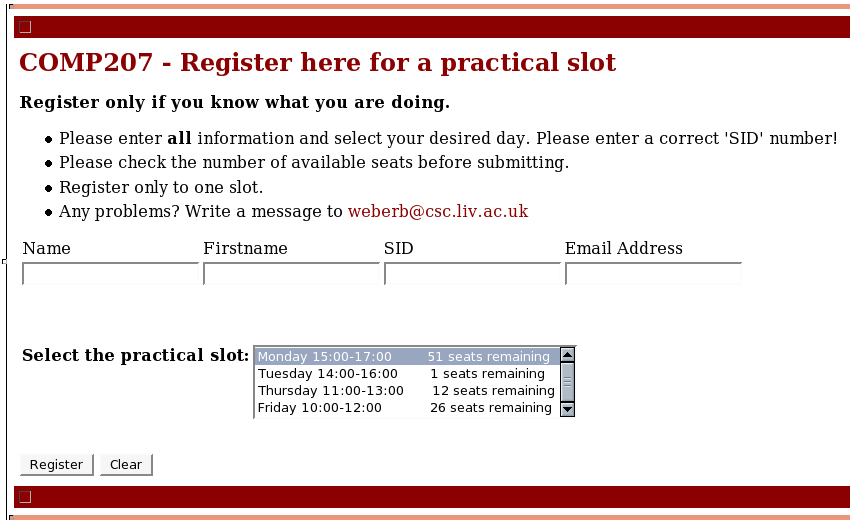
Assessment

**This assignment contributes 1/3 of the 25% continuous assessment part of the course grade. The maximum possible grade on this assignment is 100 points.** Failure in this task can be compensated by higher marks on the other assessments of the module.

## Background

You will be writing a small PHP/SQL application to deal with the following (fictionalized) problem.

There are 32 students assigned to a module. In order to organize module demonstrations, the class should be split into four groups with different time slots. Each group should consist of 8 students. The module lecturer decided to have a registration webpage to allow students to sign up for one of the demonstrations. A student visiting the page should be able to submit her name, student ID, email address, and book a place in one of the time slots. For example, the registration Web page of COMP207 looks like this:



The submitted data should be stored in a database which is maintained on a server. The webpage and the server should interact with each other at every step of the registration process. The page should show how many free places are available in each time slot, announcing and blocking all fully booked time slots. After a student makes a data submission, it should check whether the student has been already registered. If not, the data is stored on the server and the student is notified about her registration. Otherwise (if already registered), the student should be prompted to ensure that she wants to change her registration to the new section (and removed from the current one she is registered for).

As a means of checking which students are registered in each section, you should also write a **separate** webpage that will allow the module lecturer to choose a section and, after querying the database, will display the list of students who are registered in that section to a webpage.

## Description of the problem

For your assignment, you are to write a registration webpage using HTML, CSS, JavaScript, PHP and mySQL, as outlined above, in order to allow students to register to demonstration timeslots.

### Part 1 (70 pts)

The database must fully implement the registration process, maintaining the submitted information. The server and page behavior must meet the requirements listed above. In addition, (basic) help facilities should be provided. A separate page for the lecturer should also be working so that he/she can see the students currently enrolled.

### Part 2 (30 points)

The specific look and feel of the pages as well as the database implementation is left intentionally vague, allowing considerable design freedom on your part. However, the page should have a nice look and the code should satisfy common standards. You may also add additional features to your page. For example, you can add JavaScript and/or PHP functions which can validate the name (for example, that the field isn't empty), SID (that it has only numbers in it), and email fields.

**Important note**: It is assumed that your HTML/PHP code you write will comply with the XHTML Strict 1.0 standards (or better, such as HTML5). Failure to adhere to proper web coding standards will result in a loss of marks from your grade.