

# COMP1000 Sample Exam 2013s1 - Instructions

This examination has four parts:

Part 1: 30 Multiple Choice Questions (25 marks / 100)

25% of 2 hours = 30 minutes

Part 2: Practical Excel (15 marks / 100)

15% of 2 hours = 18 minutes

Part 3: Practical HTML (20 marks / 100)

20% of 2 hours = 24 minutes

Part 4: Practical Access (40 marks / 100)

40% of 2 hours = 48 minutes

## Part 1: Multiple Choice Questions

Download and open the Excel workbook `sample-exam.xlsx`. If an alert box or similar comes up asking you to update links or enable content, do so.

Answer the questions in the **Multiple Choice Questions** worksheet. **Save** and **submit** regularly.

## Part 2: Practical Excel

The Excel worksheet that you need to use is called **ResultsWAM**, and it is in the same workbook as the multiple choice questions.

- Turn on the macro recorder, and call the macro **Format**.
- Format the top row as headers: bold with a yellow background;
- Turn off the macro recorder.
- Produce a PivotTable (in a new worksheet called **PivotTable**, positioned to the right of the ResultsWAM worksheet, tab colour yellow) with *Programs* as rows, columns for *Course A* and *Course B*, recording Average values, 1 decimal place. Use suitable column headings.
- Return to the **ResultsWAM** worksheet, and select the columns headed *WAM* and *Course A*.
- Produce a scatter plot of *Course A* values (vertical axis) vs *WAM* values. Title the vertical axis *y*, the horizontal axis *x*, the plot **Course Results vs WAM**, no legend, default plot symbol, a linear trendline with equation (put the equation in the top left corner of the plot where there are no datapoints). The numbers are exam marks between 0 and 100: make sure the axis scales are appropriate. The scatter plot should cover the range E1:L20.

Before proceeding to the next part, **save** and **submit** your Excel file.

## Part 3: Practical HTML

Download the HTML template file called `flashcard.html`. Open it using Notepad (Start > All Programs > Accessories > Notepad) or an available text editor of your choice.

Make a web page based on `flashcard.html` that looks like what is shown at left below, before the Show/Hide button is clicked, but after clicking, the right pane changes to the version at right below.

### English → French Flashcard

English	French	French
sheep		mouton
Next Card	Show/Hide	Show/Hide

Notes: *mouton* is the French word for *sheep*. The boxes with *sheep* and *mouton* in them are 200×200 pixels. The `valign="center"` attribute can be used to centre the words *sheep* and *mouton* vertically (as `align="center"` does horizontally). The Next Card button implements a link to a webpage called `flashcard2.html` with the next word and its French equivalent: see the lecture notes for how to do this. The Show/Hide button alternately shows and hides the French word: see the lecture notes

for how to do this. The arrow in the heading "English → French Flashcard" is one of the HTML Character Codes - see the lecture slide with this name for how to do this. The blank window pane displays a `&nbsp;` character code. Don't forget to wrap the table in a `<form>` element to provide an environment for the buttons. Since in the exam situation you can only save & submit a single HTML file, there is no point in actually making the file `flashcard2.html`. An information sheet on HTML Forms and Javascript is provided.

Before proceeding to the next part, **save** and **submit** your HTML file.

#### *Part 4: Practical Access*

Download the Access database `prices_original.accdb`. Open it using Access. Make a copy called `prices.accdb`. If you receive a Security warning, click the Enable Content button, and click Yes to make the file a Trusted Document. You can return to `prices_original.accdb` if you mess up the copy you have just made.

##### 1. Open Table and Change Name

Open the *Employees* table, find the name Ron Lorah, replace it with your name, & close the table.

##### 2. Identify and Update Selected Category Prices

- Create a Select query that includes the CategoryID and CategoryName from the Categories table and the UnitPrice and ProductName fields from the Products table. Run the query, and then identify the CategoryID for Beverages.
- Add the appropriate criterion to limit the query output to only Beverages.
- Convert the query to an update query. Update the UnitPrice field by increasing it by 10%.
- Check in Datasheet view to ensure you are updating the right records.
- Return to Design view, and then run the query. Save the query as Update Prices.

##### 3. Create a New Table

- Create a select query that identifies all of the discontinued products.
- Convert the select query to a make table query.
- Name the new table Discontinued Products. Run the query.
- Save the query as Make Discontinued Products Table.
- Convert the make table query to a delete query. Run the query.
- Save the query as Delete Discontinued Products. Close the query.

##### 4. Calculate Summary Statistics

- Open the Profit query in Design view, and then add the LastName field from the Employees table to the design grid. Run the query, and then save the design changes. Close it.
- Use the query wizard to create a crosstab query that shows total profit by LastName and CategoryName. Name the query ProfitTable.
- Save your changes. Close the query.

##### 5. Create a query where someone can enter an employee code and see the associated orders..

- Create a query based on the Orders table. Add criteria that allow each employee to type in his or her ID and then see all of his or her orders.
- Save the query as Orders by Employee ID.
- Test the query, and then close it.

##### 6. Create a query to see if any of the employees have not processed at least one order

- Create a query to find out if any of the employees have not entered at least one order.
- Save the query as Employees With No Orders.

##### 7. Create a Stand-Alone Macro to Automatically Open Several Tables

- Your macro should open the following tables: Orders, Order Details, and Products.
- Save your macro with the name Open 3 Tables. Test your macro. Close all objects.
- Compact and repair the database.

Remember to **Save and Submit** your work regularly, and at the end of the exam! When you click **Submit** in the *final* exam, it saves files with the specified names: so be sure the names are right. Solutions to this sample exam will be available on the COMP1000 labs page at 5pm on 12 June.

## Javascript, HTML and CSS Information Sheet

Here is a list of the Javascript constructs that were covered in lectures & labs:

<code>alert( ' Put message here' )</code>	displays a Javascript message box with an OK button
<code>name.value=string   expression</code>	means "or", and name is the name of a form element, such as a text area, e.g. <code>area1.value=box2.value*1.5</code>
<code>string1 + string2</code>	concatenates strings, e.g. "Hi " + "there" = "Hi there"
<code>\n</code>	in the middle of a string means "start a new line"
<code>window.location='URL'</code>	window enclosing the form displays page at specified URL
<code>if( condition action1 ; else action2)</code>	if condition is true, do action1, otherwise do action2. E.g. <code>if( layer2.style.visibility=='hidden' )</code> <code>layer2.style.visibility='visible' ; else</code> <code>layer2.style.visibility='hidden'</code>
<code>window.open( 'URL', '_blank', height=300, width=300' )</code>	opens a new 300x300 pixel window displaying the page at the specified URL. See example at <a href="http://www.cse.unsw.edu.au/~billw/other/windowopen.html">http://www.cse.unsw.edu.au/~billw/other/windowopen.html</a>

Here is a partial list of HTML <form> elements covered in lectures, with examples of use:

button	<code>&lt;input type="button" name="btn1" value="Press me"&gt;</code>
radio button	<code>&lt;input type="radio" name="rad1" value=3&gt;</code>
check box	<code>&lt;input type="checkbox" name="chk1" value="A"&gt;</code>
text box	<code>&lt;input type="text" name="txt1" value="default text" size=20&gt;</code>
reset	<code>&lt;input type="reset" name="clear" value="Clear all input"&gt;</code>
select	<code>&lt;select&gt;&lt;option value="1"&gt;Happy&lt;/option&gt;</code> <code>    &lt;option value="2"&gt;Sad&lt;/option&gt;&lt;/select&gt;</code>
select	<code>&lt;select size="2"&gt;&lt;option value="1"&gt;Happy&lt;/option&gt;</code> <code>    &lt;option value="2"&gt;Sad&lt;/option&gt;&lt;/select&gt;</code>
textarea	<code>&lt;textarea name="area1" cols=15 rows=7&gt;Default text&lt;/textarea&gt;</code>

The syntax for a simple CSS in-line style definition is:

```
<span style="attribute: value; ">text</span>
```

where you would replace `attribute` by a CSS attribute name, `value` by the desired value for that attribute, and `text` by the text to be formatted - e.g.

```
<span style="font-size: 18pt; ">big</span> to produce big.
```

Some CSS attributes, with example values, are:

font-family: Helvetica, sans-serif ;	list-style-type: circle ;
font-style: italic ;	line-height: 1.5 ;
font-variant: small-caps ;	text-decoration: underline ;
font-weight: bold ;	text-align: center ;
font-size: 18pt ;	text-indent: 15px;
color: blue ;	background-color: yellow ;
visibility: hidden   visible ;	

HTML element syntax (some, like <a>, <p>, <br>, <h1>..<h6>, you should memorize)

<code>&lt;table border=<i>n</i> align="center" bgcolor=yellow&gt;</code>	<code>&lt;tr&gt;</code> table row
<code>&lt;th&gt;</code> table heading	<code>&lt;td rowspan=<i>m</i> colspan=<i>n</i> align="center"&gt;</code> table cell
<code>&lt;div id="name" class="mnemonic"&gt;</code> in body	<code>div.class { property list ...}</code> in <style> element in head
<code>x&lt;sup&gt;2&lt;/sup&gt; = x<sup>2</sup>, x&lt;sub&gt;2&lt;/sub&gt; = x<sub>2</sub>, &lt;hr&gt;=</code> horizontal rule, <code>&lt;img href='URL' alt="text"&gt;</code> for image	
common character codes: <code>&amp;lt;</code> = <, <code>&amp;gt;</code> = >, <code>&amp;amp;</code> = &, <code>&amp;nbsp;</code> = nonbreaking space, <code>&amp;rarr;</code> = →	