

**THE BCS PROFESSIONAL EXAMINATIONS
BCS Level 6 Professional Graduate Diploma in IT**

April 2008

EXAMINERS' REPORT

Web Engineering

General Comments

This paper supersedes *World Wide Web: Beyond the Basics* (which is now discontinued) and it was not possible to sit both examinations. This was the second time the paper was offered.

The subjects covered currently include XML, web programming, web 2.0 and other current technologies. The module attempts to be contemporary in the topics it covers, and each year will introduce some new topics within the remit of the published syllabus.

It is important for candidates to know that whilst on occasion questions may look similar to those in past papers, the context and approach is often significantly different, which means that previous answers cannot simply be restated; thus it is not appropriate to memorise and re-state past paper answers. Additionally, the answer pointers provided here give guidance and are only a guideline and should not be merely quoted by candidates, but applied to the topic of the question.

Question 1

You are acting as a consultant for a financial institution, advising on the development of an online banking service.

- (a) One possible security risk to a banking site is **brute forcing** the password; this is often solved by locking out an account after a certain number of unsuccessful attempts.

- (i) What is meant by a brute force attack?

(2 marks)

- (ii) Aside from a brute force attack, outline **FOUR** other security risks and, for each risk, describe a method to prevent it.

(8 marks)

- (b) The in-house developers have outlined three possible schemes for authenticating their users. You have been asked to comment on these systems from a *security* perspective (how safe the scheme will be) as well as from a *user's* perspective (how easy the scheme will be to use).

Proposed scheme A

- Type in your *email address*
- Type in your *password* (6 characters, alphabetic)
- Select the first and the third digits of your *PIN* (6 digits) from a drop-down list

Proposed scheme B

- System-generated *user number* (created on registration, 11 digits long)
- Type in your *password* (6-12 alphanumeric characters)
- Type in your *mother's maiden name*

Proposed scheme C

- Type in your *first name*
- Type in your *last name*
- Type in your *postcode/Zip code*
- Type in your *date of birth*
- Type in your *favourite colour*

- (i) For each of the three schemes, outline **TWO** strengths and **TWO** weaknesses.

(9 marks)

- (ii) Devise a new scheme for authenticating users on this site which is superior to all three schemes outlined above. Explain how it overcomes the weaknesses identified in part (b)(i).

(6 marks)

Answer Pointers

a)

- i) A brute force attack is a method of breaking a cryptographic scheme by trying a large number of input possibilities (e.g. trying all 10,000 possible combinations of a 4-digit PIN).
- ii) Sample risks include:
- Phishing (Microsoft's anti-phishing filter).
 - Packet sniffing (use encryption).
 - Keylogger on client machine (check physical hardware and/or run a rootkit checker/antivirus).
 - SQL injection attack (ensure all user inputs are validated on the server side).
 - Denial of service attack (maintain mirror of server application & data).
 - Spyware on client machine (use antivirus/spyware checkers).

- Not destroying letters with sensitive information e.g. passwords (shred confidential documents when disposing).

b)

i) Possible answers:

	Strengths	Weaknesses
A	<ul style="list-style-type: none"> • Reasonably easy to remember login details. • Defends against keylogging by using drop down lists • Partial defence against phishing by only needing two digits from the PIN. 	<ul style="list-style-type: none"> • Short alphabetic password gives only ~300 million combinations => brute forcible. • Requiring only the first and third digit of the PIN gives only 100 combinations.
B	<ul style="list-style-type: none"> • Reasonably strong password. • Relatively fast to log in (only three text fields to be entered). 	<ul style="list-style-type: none"> • Difficult to remember login details (randomly generate user ID), so more likely to write details down. • Mother's maiden name is known by other people (e.g. family members). • All data is typed, so susceptible to keylogging.
C	<ul style="list-style-type: none"> • Very easy to remember login details. • Requires a collection of details that would be very difficult to brute force (but not find out through other means). 	<ul style="list-style-type: none"> • Susceptible to identity fraud, as login details are almost entirely based on common personal details. • Lots of details to be typed in. • Favourite colour might be guessable (e.g. most people would choose a primary colour or black) • Susceptible to keylogging.

ii) An open question, but solutions must address all weaknesses of the previous schemes (both security and usability), whilst remaining reasonably simple for the end user. Possibilities might include:

- Selecting random digits from a 6 digit PIN instead of merely two predetermined ones.
- Asking for one personal detail from a random selection (e.g. mother's maiden name, favourite colour, and name of first pet).
- Offering to remember the system username (not advised for multi-user PCs or accessing the site from a public computer).
- Allowing passwords to be alphanumeric and of variable length, with a sensible minimum length (e.g. 6 characters).

•

Examiners' Guidance Notes

Very few candidates appeared to know what is meant by "brute forcing", with many equating it with "hacking into a server". The answers given for part a) ii) were overly generic – textbook definitions of basic threats did not carry many marks unless applied to the problem this question is concerned with – e.g. that of securing online banking. Solutions given to the problems identified were often very basic. Part b) i) was reasonably answered from a security point of view, but less so from the user's perspective. Generic statements that "the password is not secure" is not sufficiently critical to be awarded marks without a reason. Part b) ii) was poorly answered in almost every case – the idea was to demonstrate that the new scheme addresses all the weaknesses from previous schemes. Simply mentioning the use of biometrics without explaining how it addresses the problems (and without mentioning the practicalities of such schemes from a usability perspective) was not worth many marks.

Question 2

A company selling digital cameras wishes to enhance its website to incorporate e-commerce. To this end, four modules are to be added to the existing website:

- User Registration & Login;
- Shopping Cart (links into existing company financial and stock control databases);
- Check Out (converting a *Shopping Cart* into an order, taking payment); and
- Order Management (status of existing orders, cancelling pending orders, etc.).

- (a) (i) State **THREE** possible solutions to the problem of handling payments. **(3 marks)**
- (ii) Adopt **ONE** solution from part (i) and give a reason why it is preferable to each of the other solutions outlined. **(2 marks)**
- (b) In terms of the customer shopping experience on the website, outline the required functionality of the *Shopping Cart* module. **(6 marks)**
- (c) (i) Considering any **THREE** of the modules, identify the key aspects that would form a core part of an *acceptance test plan* for each. Amongst other things, the test plan should specifically address issues of performance and security.
[Note: individual detailed test cases are not required here.] **(10 marks)**
- (ii) Construct an *integration test plan* involving all four modules. **(4 marks)**

Answer Pointers

- a)
- i)
- Third party: Paypal/Google checkout
 - Credit/debit cards in real time
 - Credit/debit cards over the phone
 - Cash on Delivery
 - Vouchers
 - Cheques
- ii) Any logically argued and consistent method is acceptable, so long as it considers speed, security, and cost. The argument should consider the relative size of the company, number of transactions per time period, etc.
The recommendation of slow methods (e.g. cheques, COD) would have to have a very strong justification.
Depending on the cost of custom built check out with the added cost of security and fraud prevention (Authentication, Authorisation and settlement), it might be cheaper to adopt one of the following on offer say from Google
- Buy Now buttons
 - Email invoicing
 - Off-the-shelf shopping trolleys
 - Custom-built shopping trolleys

b)

Functionality could include:

Browse items, search for an item, place an order for a single item, place an order for a number of items, Check for availability of an item, amend an order to increase/decrease the quantity as well as deleting an item. Have an itemised cost and running total and postage/delivery costs, times including taxes, save shopping cart for later.

c)

i) Construct a coherent unit test strategy looking only at the individual modules, not at the interaction. Possible aspects to consider could include:

User Registration and login	Shopping Cart	Check out	Order Management
Load testing: <ul style="list-style-type: none"> No of concurrent users Verification of functionality: <ul style="list-style-type: none"> Address lookup on post code Data Validation: <ul style="list-style-type: none"> House number/name Email address – valid characters and verification Title (Mr, Mrs, etc.) First name & Last name – valid characters (accented letters?) Password – a mix of characters and digits and a min length Unique username 	Load testing: <ul style="list-style-type: none"> No of concurrent users Verification of functionality: <ul style="list-style-type: none"> Browse item to display correct amount in stock Order a single item Order multiple items Amend orders to increase/decrease quantity by one, qty 1 to 0 should be removed from the cart Qty cannot exceed stock level Qty must be positive digits with error trapping and offer to re-enter qty Each item selected should be displayed with a qty chosen and cost and running total Check the figures On-the-fly currency conversion estimation? 	Load testing: <ul style="list-style-type: none"> No of concurrent users Verification of functionality: <ul style="list-style-type: none"> Currency conversion/choice of currencies? Payment options Correct delivery charges Correct taxes Delivery address Delivery options – free express, next day Order saved for a latter session Confirmation of order Confirmation of payment approval Verification of security: <ul style="list-style-type: none"> Valid server digital certificate SSL, appropriate 	Load testing: <ul style="list-style-type: none"> No of concurrent users Verification of functionality: <ul style="list-style-type: none"> Correct Retrieval of an order Time stamped for cancellation to be valid until Payment not approved – order cancelled or user informed Time until which order can be amended – amendments to be done – confirmation of amended order

ii) The integration test should consider the interaction between the different modules – e.g. adding products to the shopping cart and then checking out, trying to check out when not logged in, cancelling an order that doesn't exist, adding items to a shopping cart when not logged in, trying to check out with an empty shopping cart, etc.

Examiners' Guidance Notes

Part a) was well answered. Part b) was considering the site from the user's perspective, but many candidates simply quoted the answer pointers from last year's paper – which did not apply this year. For example, candidates would talk about the “producer” and “owner” of the site, instead of focussing solely on the “user.” Part c) was poorly answered. Candidates failed to appreciate that the acceptance test would have been conducted by the owner, not the users. Part c) ii) was answered well in some cases, but many were quite theoretical in nature rather than considering the actual modules outlined in the question.

Question 3

- (a) Write HTML to construct the form as indicated in Figure 3.1. When the button marked *Create Account* is pressed, the form should submit to a script called `create.php`
(3 marks)
- (b) (i) What are the implications of *not* validating data?
(2 marks)
- (ii) Describe, with an example for each, when data should be validated at:
- the client end; and
 - the server end.
- (3 marks)
- (c) Write code (either at the client side in JavaScript, or at the server side in ASP, PHP, or Perl) to validate that:
- (i) The `username` field exists and consists of 6 alphanumeric characters only.
(3 marks)
- (ii) The `mobilephone` field is in the format “(07xxx) xxxxxx” where each “x” represents a single digit.
(2 marks)
- (iii) The email field is in the format “<name>@<server>.<location>” (e.g. `invigilator-exams.room_5@bcs.org.uk`) where:
- (1) <name> consists of a combination of one or more alphanumeric characters, full stops (.), hyphens (-) and/or underscores (_), e.g. “invigilator-exams.room_5”.
 - (2) <server> is alphabetic only, e.g. “bcs”.
 - (3) <location> consists of a combination of one or more alphanumeric characters and/or full stops (.) e.g. “org.uk” or “com”.
- The code should return an appropriate and informative error message if the validation fails.
(3 marks)
- (d) The server has a database named **ACCOUNTS**, with a single table named **User** (as shown in Figure 3.2). You may assume the web server and the database server are the same machine.
- For parts (ii) and (iii) below, assume that the script `create.php` has already established a valid connection to this database. In all cases, you should state the language you are using (ASP, PHP or Perl).
- (i) Write code to connect to the database.
(1 mark)

- (ii) Write code to store the details submitted from the form into the database. **(3 marks)**
- (iii) Write code to retrieve and display (in an appropriate format) the details of *all* accounts in the database which have an email address ending in “.co.uk”. **(5 marks)**

Note: The following SQL syntax may be useful to accomplish these tasks:

- `INSERT INTO tbl_name (col1, col2, ...) VALUES (val1, val2, ...);`
- `SELECT * FROM tbl_name WHERE col1 like "%val1%";`

(Where tbl_name, col1, val1 etc. are to be replaced with appropriate values.)

Figure 3.1: HTML form

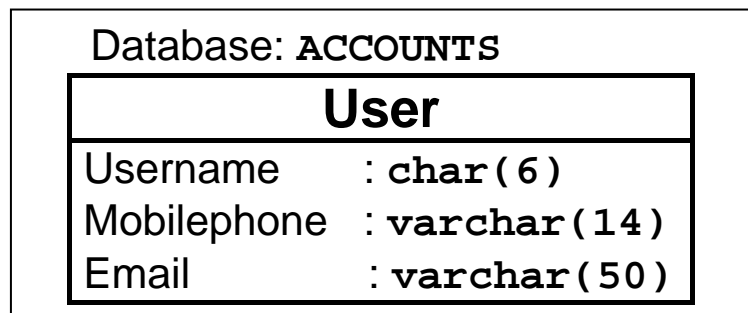


Figure 3.2: Database Entity-Attribute diagram

Answer Pointers

a)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
    <title>Create Account</title>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">
</head>
<body>
<h1>Create new Account</h1>
    <form action="password.php" method="get">
        <p>Enter a new username (6 characters): <input type="text"
        value="fred99" name="username">
        <br>Enter your mobile phone number: <input type="text"
        value="(07999) 999999" name="mobilephone">
        <br>Enter your email address: <input type="text"
        value="a@a.invalid" name="email">
        <br><input type="submit" value="Create Account">
        <input type="reset" value="Clear Form">
    </p>
    </form>
</body>
</html>
```

b)

- i) For a program to operate effectively, correct data is vital. Data validation is important because it is not possible to predict what the user agent will enter into input fields. Data may be corrupt, missing, badly formed or even maliciously designed to exploit weaknesses in the program.
Data should be validated at client end when it is important to have quick feedback to the user of any errors e.g. forgetting to fill in a postcode.
- ii) Whilst it is true that all important data should be validated at server end (regardless of the presence of client-side validation), and this should be given partial credit, an answer for full marks should consider a case where client side validation is not feasible, and validation must take place at the server end. E.g. if it relies on checking the data using some external means (e.g. a Postcode against the Royal Mail database).

c)

i)

```
// where 'username' is the name of the textfield defined in part a)
$username = $_GET['username'];
$pattern = '/^[a-zA-Z\d]{6}$/';
$matches = "";
if(!preg_match($pattern, $username))
{
    echo "<p>User name must contain 6 alphanumeric characters only.</p>\n";
} else {
    // SOME CODE
}
```

ii)

```
// where 'mobilephone' is the name of the textfield defined in part a)
$username = $_GET['mobilephone'];
```



```

$pattern = '/^\(07[\d]{3}\) [\d]{6}$/' ;
$matches = "";
if(!preg_match($pattern, $username))
{
    echo "<p>Mobile phone number must be in the format (07xxx) xxxxxx, where x
        represent a digit.</p>\n";
} else {
    // SOME CODE
}

iii)
// where 'email' is the name of the textfield defined in part a)
$email = $_GET['email'];
$pattern = '/^[a-zA-Z\._-]+@[a-zA-Z]+\.[\w\.]+'$/';
$matches = "";
if(!preg_match($pattern, $email))
{
    echo "<p>Email address is not in a valid format.</p>\n";
} else {
    // SOME CODE
}

```

d)

```

i)
$link = mysql_connect('localhost:3307', 'user', 'password');
(Note: mysql_connect and server name localhost or 127.0.0.1 are required)

```

```

ii)
// assume connection already established from part d) i)
$username = $_GET['username'];
$mobilephone = $_GET['mobilephone'];
$email = $_GET['email'];
$username = addslashes($username);
$mobilephone = addslashes($mobilephone);
$email = addslashes($email);
$result = mysql_select_db("ACCOUNTS", $link);
if (! $result) {
    echo "Failed to connect to database.\n";
}
else
{
    $result = mysql_query("insert into User (username, mobilephone, email)
        values ('$username', '$mobilephone', '$email');", $link);

    if (! $result) {
        echo "<p>MySQL Error: " . mysql_error($link) . "</p>\n";
    }
    else
    {
        echo "<p>User " . $username . " added to the database.</p>";
    }
}
mysql_close($link);

```

```

iii)
// assume connection already established from part d) i)
$key = ".co.uk";
$result = mysql_select_db("SECURITY", $link);
if (! $result) {
    echo "Failed to connect to database.\n";
}
else
{
    $result = mysql_query("select Username from User where email like '%$key%',
        $link);
    if (! $result) {
        echo "<p>MySQL Error: " . mysql_error($link) . "</p>\n";
    }
    else
    {
        $all_results = mysql_fetch_array($result);
        if(empty($all_results))
        {
            echo "<p>No users have an email address ending in: " . $key . "</p>\n";
        }
        else
        {
            echo "<table>\n";
            echo "  <tr><th>Username</th></tr>\n";
            foreach($all_results as $username)
            {
                echo "  <tr><td>$username</td></tr>\n";
            }
            echo "</table>\n";
        }
    }
}
mysql_close($link);

```

Examiners' Guidance Notes

Part a) was generally well answered, but candidates should remember to explicitly state the programming language they are using (from PHP, ASP or JSP). Part b) i) was poorly answered – candidates gave reasons why we should validate data, but many did not consider the question actually posed – i.e. what could happen if you do not validate data. Stronger students indicated realistic risks. Part b) ii) was answered well by some students, but many students simply quoted answer pointers for a similar past question and entirely missed the “with an example for each” element. Part c) was not answered by a sizeable proportion of candidates that attempted this question. Part c) i) was generally well answered by those that attempted this part. Part c) ii) and iii) were only answered by a few students, and typically at a very superficial level. Answers varied between being over-complex and over-simplistic with the data validation checks requested. Part d) was generally well answered.

Question 4

- (a)
- (i) List the characteristics of **static** and **dynamic** web pages. (3 marks)
 - (ii) Outline the technologies and tools used in the creation of a static web page containing images. (3 marks)
 - (iii) Aside from those elements mentioned in (ii) above, outline the additional technologies and tools used in the creation of a dynamic web page. (4 marks)
- (b) Figure 4.3 (overleaf) details the source code of a web site for an electronics retailer.
- (i) Draw a diagram to complete the missing sections A, B and C indicated in Figure 4.1 (below) to illustrate the output of this file when it is first loaded in a browser window. (State the browser you are assuming use of.)(3 marks)
 - (ii) The page has JavaScript actions attached to it. Describe what will happen on screen in relation to user interaction with the page. (4 marks)
 - (iii) The retailer wishes to add extra details to the web page, with the same format and functionality as the current content. Write code to enable the menu group displayed in Figure 4.2 (below) to be generated and displayed in the same way as the other groups. (4 marks)
 - (iv) Using CSS, list modifications to the code to change the appearance of the listed groups to the following:
Cameras: Bold text, surrounded by a solid 3-pixel border (2 marks)
PCs: Italic text, surrounded by a dotted 2-pixel border (2 marks)
(Note: Your answer should list only the changes and additional lines required.)

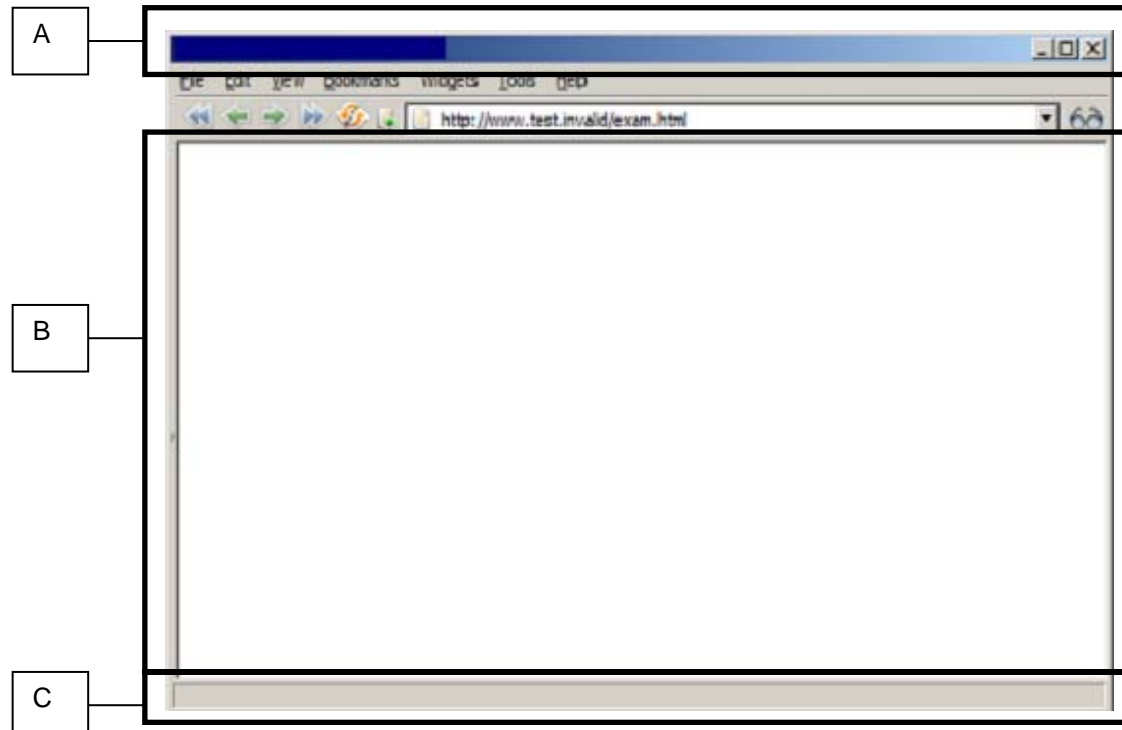


Figure 4.1: Template browser page for question 4

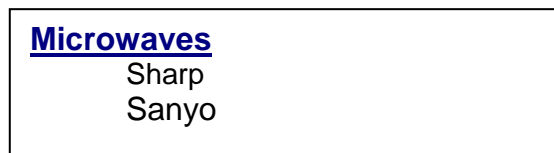


Figure 4.2: Menu group to add for question 4

```

1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3: <html>
4:     <head>
5:         <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
6:         <title>Electronics Limited</title>
7:         <script type="text/javascript" language="javascript">
8:             <!--
9:             var activegroup = 0;
10:            var mingroupnum = 1;
11:            var maxgroupnum = 3;
12:
13:            function togglegroup(){
14:                activegroup++;
15:                if(activegroup > maxgroupnum) {
16:                    activegroup = mingroupnum;
17:                }
18:
19:            for(i = mingroupnum; i <= maxgroupnum; i++) {
20:                thisgroup = eval("document.all.group"+i+".style");
21:
22:                if(i == activegroup){
23:                    thisgroup.display = "block";
24:                }
25:                else {
26:                    thisgroup.display = "none";
27:                }
28:            }
29:        }
30:        -->
31:    </script>
32:    <style type="text/css">
33:        .group { display:none; margin-left:20px;}
34:    </style>
35: </head>
36: <body bgcolor="white">
37:     <h1 onmousedown="togglegroup()">Click here:</h1>
38:     <h3><a href="page1.html">
39:         Cameras</a></h3>
40:     <p id="group1" class="group">
41:         Sony<br />
42:         Canon
43:     </p>
44:     <h3><a href="page2.html">
45:         Televisions</a></h3>
46:     <p id="group2" class="group">
47:         Panasonic<br />
48:         Samsung
49:     </p>
50:     <h3><a href="page3.html">
51:         PCs</a></h3>
52:     <p id="group3" class="group">
53:         Dell<br />
54:         IBM
55:     </p>
56:     <p>
57:         <a href="http://validator.w3.org/check?uri=referer"></a>
60:     </p>
61: </body>
62: </html>

```

Figure 4.3: Source code for question 4

Answer Pointers

a)

i) A static web page has the following characteristics:

- Interaction only through traversing hyperlinks or through plugins.
- Information controlled by the website creator.
- No ability to configure the presentational aspects of the site, except at a local browser level.

A dynamic page tends to have at least some of the following characteristics:

- Can respond to input data provided by the web browser when retrieving the page.
- Can alter or customise the output based on certain factors or criteria.
- Can obtain information from external sources e.g. databases.
- Can be transaction based

ii)

Static development tools and technologies include:

- Web page editor (either a standard text editor e.g. Notepad, or a WYSIWYG editor, e.g. Dreamweaver) for writing HTML.
- Image acquisition and manipulation tools e.g. Scanner, Photoshop
- File transfer software (such as FTP - possibly built into the web page editor)
- Web server (e.g. Apache)

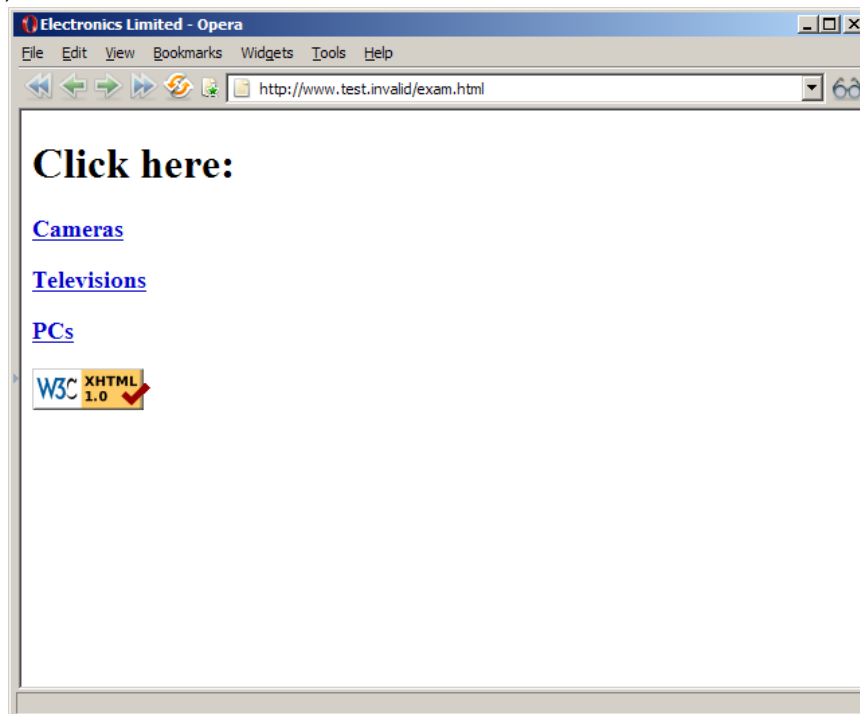
iii)

Dynamic development tools and technologies include:

- Scripting languages (e.g. PHP, ASP, JSP, Coldfusion, Perl)
- Possibly development tools/IDE for the dynamic language (e.g. Coldfusion, NetBeans).
- Web server supporting the chosen dynamic languages

b)

i)



When the mouse is clicked over the page header (the text marked “click here:”, the first menu will open. When it is clicked a subsequent time, it will close the current menu and open the next one (e.g. from Cameras to televisions, from televisions to PCs, from PCs to cameras).

ii)

Amend line 11:

```
var maxgroupnum = 4;
```

Add in line 50:

```
<h3><a href="page4.html">
    Microwaves</a></h3>
<p id="group4" class="group">
    Sharp<br />
    Sanyo
</p>
```

iii)

Add the following code after line 33:

```
.camera { font-weight: bold; border: 3px solid black; }
.pc      { font-style: italic; border: 2px dotted black; }
```

Amend line 40:

```
<p id="group1" class="group camera">
```

Amend line 52:

```
<p id="group2" class="group pc">
```

Examiners' Guidance Notes

In general, this question was answered well. However, part b) ii) was poorly answered – candidates assumed that it was the question from a previous year, but the JavaScript actions were entirely different – as they are on each paper.

Question 5

(a) A simple DTD for an academic journal is provided in figure 5.1.

(i) Explain what it means for an XML document to be *well formed* and *valid*.

(2 marks)

(ii) The XML document in figure 5.2 contains precisely **FOUR** errors when validated against the DTD in figure 5.1. Identify all the errors, and provide a solution for each one.

[Note: the line numbers are for your benefit and are not part of the XML code.]

(4 marks)

(iii) The original DTD has been found to be too simplistic for practical use. Referring to the XML document in figure 5.3, write a DTD which **enforces** the following **ten** constraints on JournalCatalogue:

- Sequence of elements is as shown in the XML code above (i.e. JournalCatalogue is a container of Journal elements).
- The two attributes in Journal are mandatory.
- Journal_url and Editor_email are optional.
- Journal_url must have a page attribute.
- For each Journal, exactly one Journal_title and one Editor element must be present.
- For each Journal, the Article element must be present one or more times.

- For each Article, exactly one Article_title and one Abstract must be present and the Author element must be present one or more times.
- An Author has a Name and optionally an Occupation.
- A Name consists of both a First_name and a Last_name.

(9 marks)

(b) The website designers wish the XML list of journals to be displayed as a web page as indicated in figure 5.4. Using the HTML template provided in figure 5.5, complete the missing code in figure 5.5 (the section marked <!-- TO BE COMPLETED -->) to accomplish this.

(10 marks)

```

1 : <?xml version = "1.0" encoding="ISO-8859-1"?>
2 : <!ELEMENT JournalCatalogue (Journal*)>
3 : <!ELEMENT Journal (Journal_title,Journal_url)>
4 : <!ATTLIST Journal ISBN CDATA "0-000-0000-0">
5 : <!ELEMENT Journal_title (#PCDATA)>
6 : <!ELEMENT Journal_url EMPTY>
7 : <!ATTLIST Journal_url page CDATA #REQUIRED>

```

Figure 5.1: XML simple DTD

```

1 : <?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
2 :     <!DOCTYPE JournalCatalogue SYSTEM "journal-initial.dtd">
3 :     <?xml-stylesheet type="text/xsl" href="journal.xsl"?>
4 : <JournalCatalogue>
5 :     <Journal ISBN="0-123-4567-1">
6 :         <Journal_Title>Nature</Journal_Title>
7 :         <Journal_url>
8 :     </Journal>
9 : </JournalCatalogue>

```

Figure 5.2: XML document with errors

```

1 : <?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
2 :     <!DOCTYPE JournalCatalogue SYSTEM "journal-ans.dtd">
3 :     <?xml-stylesheet type="text/xsl" href="journal-Cii.xsl"?>
4 : <JournalCatalogue>
5 :     <Journal ISBN="10104556" Issue_no="4">
6 :         <Journal_title>Nature</Journal_title>
7 :         <Journal_url page="http://www.nature.co.uk" />
8 :         <Editor>Bob Jones</Editor>
9 :         <Editor_email>bjones@whitensonz.com</Editor_email>
10:        <Article>
11:            <Article_title>Go green</Article_title>
12:            <Abstract>Global warming and eco systems</Abstract>
13:            <Author>
14:                <Name>
15:                    <Last_name>Midori</Last_name>
16:                    <First_name>Vert</First_name>
17:                </Name>
18:                <Occupation>Academic</Occupation>
19:            </Author>
20:        </Article>
21:        <Article>
22:            <Article_title>See red</Article_title>
23:            <Abstract>Eco activism in developed countries</Abstract>
24:            <Author>
25:                <Name>
26:                    <Last_name>Akai</Last_name>
27:                    <First_name>Rouge</First_name>
28:                </Name>
29:                <Occupation>Academic</Occupation>
30:            </Author>
31:        </Article>
32:    </Journal>
33:    <Journal ISBN="10104557" Issue_no="7">
34:        <Journal_title>Technology Today</Journal_title>
35:        <Editor>Michael Jackson</Editor>
36:        <Editor_email>jackmic@conv.com</Editor_email>
37:        <Article>
38:            <Article_title>Convergence</Article_title>
39:            <Abstract>One device as a container for many devices</Abstract>
40:            <Author>
41:                <Name>
42:                    <Last_name>Gates</Last_name>
43:                    <First_name>Billy</First_name>
44:                </Name>
45:                <Occupation>Engineer</Occupation>
46:            </Author>
47:            <Author>
48:                <Name>
49:                    <Last_name>Careers</Last_name>
50:                    <First_name>Steph</First_name>
51:                </Name>
52:                <Occupation>Academic</Occupation>
53:            </Author>
54:        </Article>
55:    </Journal>
56: </JournalCatalogue>

```

Figure 5.3: XML sample document

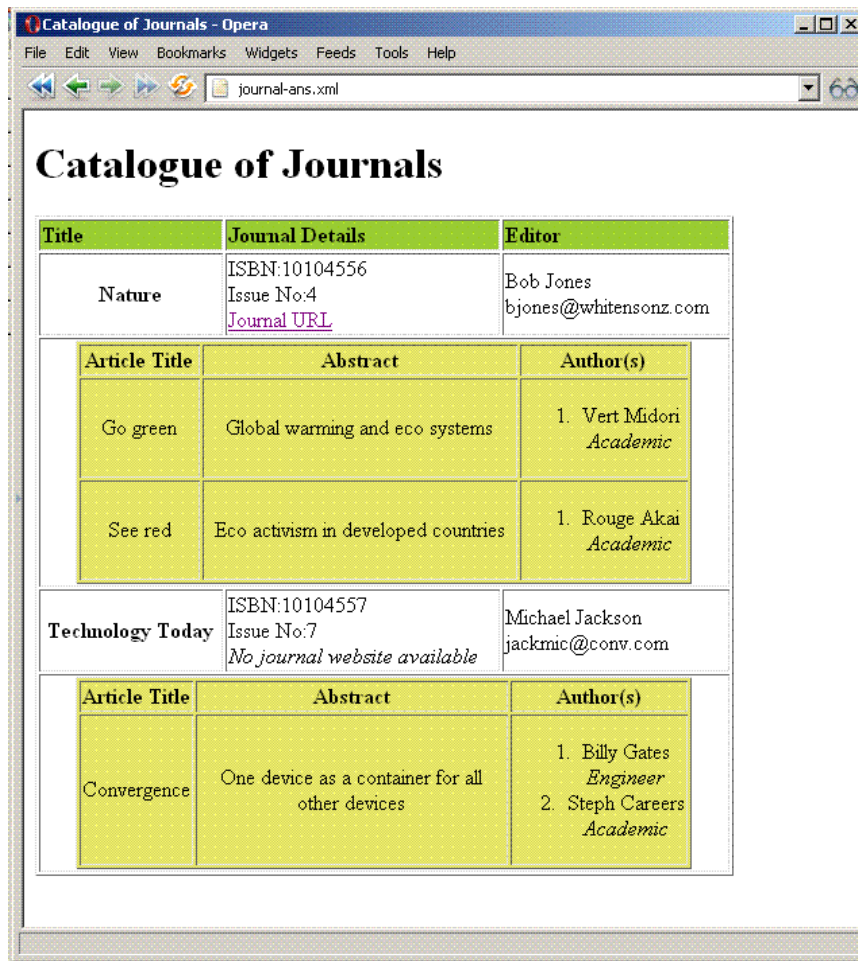


Figure 5.4: Sample browser page

Figure 5.5: Template HTML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Catalogue of Journals</title>
      </head>
      <body>
        <h1>Catalogue of Journals</h1>
        <table border="1">
          <tr bgcolor="#9acd32">
            <th align="left">Title</th>
            <th align="left">Journal Details</th>
            <th align="left">Editor</th>
          </tr>

          <!--TO BE COMPLETED -->

        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Answer Pointers

a)

i)

- A well-formed document conforms to all XML syntax rules (e.g. opening tags must all have a closing tag, be nested properly etc.).
- A valid document is a well-formed XML document that conforms to semantic rules (as defined in a DTD or XML schema)

ii) Errors are:

- Line 5: Attribute undefined (IBSN should be ISBN)
- Line 6: Undefined element type (Journal_Title should be Journal_title)
- Line 7: No end tag for Journal_url
- Line 7: Required attribute not specified (Jounal_url)

iii)

```
<?xml version = "1.0" encoding="ISO-8859-1"?>
<!--ELEMENT JournalCatalogue (Journal*)-->
<!--ELEMENT Journal (Journal_title,Journal_url?,Editor,Editor_email?,Article+)-->
<!--ATTLIST Journal ISBN CDATA #REQUIRED Issue_no CDATA #REQUIRED-->
<!--ELEMENT Journal_title (#PCDATA)-->
<!--ELEMENT Journal_url EMPTY-->
<!--ATTLIST Journal_url page CDATA #REQUIRED-->
<!--ELEMENT Editor (#PCDATA)-->
<!--ELEMENT Editor_email (#PCDATA)-->
<!--ELEMENT Article (Article_title,Abstract,Author+)-->
<!--ELEMENT Article_title (#PCDATA)-->
<!--ELEMENT Abstract (#PCDATA)-->
<!--ELEMENT Author (Name,Occupation?)-->
<!--ELEMENT Name (Last_name,First_name)-->
<!--ELEMENT Last_name (#PCDATA)-->
<!--ELEMENT First_name (#PCDATA)-->
<!--ELEMENT Occupation (#PCDATA)-->
```

b)

```
<xsl:for-each select="JournalCatalogue/Journal">
  <tr>
    <th><xsl:value-of select="Journal_title"/></th>
    <td>ISBN:<xsl:value-of select="@ISBN"/><br />
    Issue No:<xsl:value-of select="@Issue_no"/><br />
    <xsl:choose>
      <xsl:when test="Journal_url/@page">
        URL: <xsl:value-of select="Journal_url/@page "/><br />
      </xsl:when>
      <xsl:otherwise><em>No journal website available</em></xsl:otherwise>
    </xsl:choose>
    </td>
    <td><xsl:value-of select="Editor"/><br />
    <xsl:value-of select="Editor_email"/></td>
  </tr>
  <tr>
    <td colspan="3" align="center">
      <table border="1" bgcolor="#e7e7e7" width="90%">
        <tr><th>Article Title</th><th>Abstract</th><th>Author(s)</th></tr>
        <xsl:for-each select="Article">
          <tr>
            <td><xsl:value-of select="Article_title"/></td>
            <td><xsl:value-of select="Abstract"/></td>
            <td>
              <ol>
                <xsl:for-each select="Author">
                  <li><xsl:value-of select="Name/First_name"/>&#xa0;<xsl:value-of
                    select="Name/Last_name"/>
                    <br /><em><xsl:value-of select="Occupation"/></em>
                  </li>
                </xsl:for-each>
              </ol>
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </td>
  </tr>
</xsl:for-each>
```

Examiners' Guidance Notes

Part a) was generally well answered. In part b) the majority of those who attempted the question managed to get an outline solution for partial credit, however the nested and conditional processing proved to be a great challenge, with only a handful of candidates successfully providing a full and correct answer. Some candidates simply wrote HTML code to output the sample page, which gained zero marks – the question was not about HTML, but about demonstrating knowledge of XSLT.