

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS BCS Level 6 Professional Graduate Diploma in IT

WEB ENGINEERING

EXAMINERS' REPORT

General Comments

There is a large disparity between candidate answers. Candidates should remember that the answer pointers provided here give guidance and are only a guideline. They should not be quoted in answers but applied to the topic of the question.

When developing an argument (e.g. discussing the benefits or drawbacks of a certain approach), candidates should ensure they fully understand the meaning of the technical terms they are using. In particular, "cost-effective" repeatedly appeared in a number of answers this year but used in a manner which suggested the candidate meant "costly".

Candidates should also be aware that questions require the application of knowledge rather than just a restatement of that knowledge.

It would appear that some candidates would benefit from revision for questions which require coding.

A1. Some people believe that web services should not require users to log in or authenticate in any way and should instead be open for all to use.

- a) i) With a suitable example, briefly explain why **authentication** might be necessary.

(2 marks)

- ii) One approach to authentication is the use of a *username* and *password*.

A recent analysis indicated that many users had an easily guessed password such as "123456" or "password". To prevent this, companies often have password policies to ensure that users do not choose passwords that are easily broken by a dictionary attack, like the three schemes that follow:

- *Scheme A*: Password must have at least one letter and one number
- *Scheme B*: Password must have exactly 8 letters and not be based on a dictionary word.
- *Scheme C*: Password must have more than 16 letters and contain at least one number and non-alphanumeric character.

For **each** scheme, identify ONE weakness from either a security or usability perspective.

(3 marks)

iii) **State** and **justify** which of these is the most appropriate to use for a public message board system.

(2 mark)

b) Another key security threat to password-based authentication is a brute-force attack on passwords.

i) With specific details on how this differs from a dictionary attack, explain the **brute-force attack**.

(2 marks)

ii) To defend against brute forcing, services often **lock out** accounts if too many incorrect attempts to log in are made. Explain how – despite the existence of a “lock out” mechanism – an attacker might be able to brute force user passwords.

(2 marks)

c) Passwords are typically stored on the Server as a hash value, rather than as plaintext.

i) With reference to a suitable example cryptographic technique, describe the difference between **plaintext** and **hash value**.

(2 marks)

ii) In the context of hashing, define the term **salt** and explain why it should be used.

(2 marks)

iii) A company has some sensitive pages and feels that password protection alone is not enough. Your manager would like to understand some options.

Briefly describe the following two methods of authentication, and state ONE **disadvantage** of using each method.

- a. 2-factor authentication using a token-generating device (e.g. RSA SecurID);
- b. Fingerprint verification.

(4 marks)

d) Politicians have often called for the banning of encryption, for Law-enforcement reasons (e.g. intercepting communications between terrorists)

Considering the technological and ethical factors, **discuss** the statement “*all encryption should be banned.*”

You should ensure that you present a balanced argument (considering situations when banning encryption might have positive and negative consequences) and come to a clear conclusion.

(6 marks)

Answer Pointers

a) i) Authentication may be required when storing sensitive information, or personal settings that the service may rely on, but that you would not want other's accessing (e.g. financial or medical data), or for the use of administration functions.

[1 mark for an appropriate reason, 1 mark for a strong example]

ii)

Scheme A	Security: No minimum length means that a simple password like "a1" would be acceptable. This is easily brute forced.
Scheme B	Security: Fixed length of password limits the maximum strength of the password. Usability: Fixed length of password limits the opportunity of using a "passphrase" and so may mean passwords are harder to remember.
Scheme C	The requirement for a very long and complex password will make this difficult for a user to remember, which makes it more likely they will write it down.

[1 mark for each scheme]

iii) An argument can be made for any of the three options, but Candidates are most likely to select scheme B, as a good compromise. A message board system is unlikely to have any sensitive data on the user, so the consequence of a broken password is less severe, so the usability issues of Scheme 3 are likely to make it unsuitable, but a candidate with a convincing argument should not be penalised.

b) i) Brute forcing is the technique of systematically trying all possible combinations until the correct one is found. [1 mark] The length of time this takes is proportional to the length of the passphrase (in the order of M^n , where n is the length of the passphrase, and M is the total number of possible symbols that can be used for each character). This differs from a dictionary attack which instead tries variations and combinations of common dictionary words. [1 mark for either clarifying statement, or similar]

ii) If the file listing all encrypted passwords is obtained by an attacker, and the algorithm known [1 mark], they can run automated tools on a local device to break the password, and only try on the live service when they have the correct combination. [1 mark]

c) i) Candidates should describe these in their own words: [1 mark]

"A hash value is a numeric value of a fixed length that uniquely identifies data. Hash values represent large amounts of data as much smaller numeric values, so they are used with digital signatures. You can sign a hash value more efficiently than signing the larger value." – "Ensuring Data Integrity with Hash Codes, docs.microsoft.com/security/..., last accessed 25th October 2017

An example cryptographic technique is the obsolete MD5 function which reduces any size plaintext to a 128-bit hash value e.g. "hello" becomes

5d41402abc4b2a76b9719d9110117c592 (in hexadecimal). Any common example would be suitable. [1 mark]

ii) Salt values are random data used as a second input (along with a password) for a hash function. [1 mark]

Salt is used to mitigate the risk of a dictionary attack or use of a “rainbow table” (a table of precomputed hashes for passwords) to look up the password that generates a known hash. [1 mark]

iii)

2-factor authentication using a token-generating device	2FA is a method of authentication requiring not only a username/password, but something that the user has on them that other people cannot have access to. A key-generating device makes a time-limited code that must be entered each time the user logs on, and typically changes extremely rapidly (every few minutes at the most).	There will be a cost attached to issuing a physical device, and the system would not be usable if the item is not present.
Fingerprint verification	A method of authentication by pattern matching some characteristic captured from a user’s fingerprint and comparing it to that captured from a previous time.	Requires storage of personal data, and if compromised biometrics cannot be changed.

[1 mark for each description and 1 mark for a relevant drawback]

d) This is a discussion question. Marks will be allocated for:

- The depth of discussion on each of the two sides of the discussion to be considered (i.e. positive and negative aspects) [2 marks for each aspect, based on quality of argument and supporting examples]
- The structure and completeness of the discussion overall [1 mark]
- Stating a definite answer to the question [1 mark]; this is marked on the quality of this conclusion and degree to which is supported by the main argument.

A1. Examiners’ Guidance Notes

This was a popular question with a high pass rate.

There is evidence that most candidates were able to answer part (a) and (b). Some candidates answered part (a.iii) by selecting “Schema B” with a convincing argument,

some others were selecting either schema a or c where some of them gave good and convincing arguments. Most candidates struggled to define the term “salt”.

Some candidates did well in part (d) and explained positive and negative consequences, whereas others made partial answers which focused more on one element of the question (i.e., either negatives or positives).

A2 You are developing a simple web-based application to work out whether people are old enough to watch a film showing in the cinema.

N.B. This question involves server-side scripting. The preferred language is PHP but other languages are also accepted. **Clearly state which server-side scripting language you will be using for the whole question**, and make sure all relevant files are named accordingly.

a) Write HTML to construct the form as indicated in figure A2.1. When the button marked “Search for films” is pressed, the form should submit to a script called `films.php` (suitably amended if you are using a different language).

(4 marks)

b) When invalid data is entered into a form, a well-designed system will spot the invalid data and report them back to the user with an error message.

i) Why is it important to catch invalid data?

(2 marks)

ii) What would be an appropriate error message for a login attempt that fails due to an incorrect password (but correct username)?

(2 marks)

c) Better systems will automatically regenerate the form with most data already completed (to save the user from entering it a second time).

i) With a justification, give one example of form data that should NOT be automatically completed in a regenerated form.

(2 marks)

ii) Write code (at the server side in your preferred language) to validate that the date of birth entered is not in the future, and if there is an error regenerate the original form from part a) with the entered data already filled in.

(5 marks)

Notes:

- You do not require regular expressions to complete this task, but you may use them if you wish.
- You may find e.g. the PHP function `int strtotime (string $time)` useful in this task (which parses a string into a Unix timestamp), where `$time` is a textual description of a date/time e.g.

"2017-12-31" for the 31st December, 2017 or "now" for the current system time on execution.

d) The server has a database named **Films**, with a single table named **FilmRatings** (as shown in figure A2.2). You may assume the web server and the database server are run on the same machine. This database holds details of age ratings for films which are listed in figure A2.3.

For parts ii) and iii), assume that the script `films.php` (or similar) has already established a valid connection to this database.

In all cases, you should state the language you are using.

i) Write code to connect to the database.

(1 mark)

ii) Write code to retrieve the details of the film with the title that is exactly the same as the text entered in the "Enter the film title" field.

(3 marks)

iii) Write code to retrieve and display (in an appropriate format) the details of *all* films which can be bought by the person whose date of birth has been entered.

(6 marks)

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

- `SELECT * FROM tbl_name WHERE coll = vall;`

(Where `tbl_name`, `coll`, `vall` etc. are to be replaced with appropriate values)

Answer Pointers

a)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
  <title>Film Rating Search</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body>
<h1>Search Films</h1>
  <form action="films.php" method="get">
    <p>Enter the film title:
      <input type="text" value="The Wizard of Oz"
name="filmname"></p>
    <p>Enter your date of birth:
      <input type="text" value="14" name="day"
maxlength="2">
      <input type="text" value="7" name="month"
maxlength="2">
```

```

        <input type="text" value="1989" name="year"
maxlength="4"></p>
        <p>
            <input type="submit" value="Search for films">
            <input type="reset" value="Clear Form">
        </p>
    </form>
</body>
</html>

```

(0.5 mark for correct header including page title, 1 mark for correct <h1> and <p> text, 0.5 marks for correct form setup (including action), 1.5 marks for the correct input fields, 0.5 mark for correct submit field.

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.

b) ii) Something in the vein of "Invalid username/password combination – please re-check your details and try again". 0.5 marks should be given for a message such as "Invalid password" (since it gives information that the username might have been correct).

c) i) An example of data that should not be regenerated would be passwords - this is because a) the form data may not be transmitted using HTTPS and b) because the password will be visible in the source code of the cached page (which is a potential security risk).

c) ii)

```

$dateOfBirth = $year . "-" . $month . "-" . $day;
$ageInSeconds = strtotime("now") -
strtotime($dateOfBirth);
if $ageInSeconds < 0 {
?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
    <head>
        <title>Film Rating Search</title>
        <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
    </head>
    <body>
        <h1>Search Flms</h1>
        <form action="films.php" method="get">
            <p>Enter the film title:
                <input type="text" value="<? echo
$title; ?>" name="filmname">
            <br>Enter your date of birth:
                *<input type="text" size="2" value="<? echo
$day; ?>" name="day">

```

```

        *<input type="text" size="2" value="<? echo
$month; ?>" " name="month">
        *<input type="text" size="4" value="<? echo
$year; ?>" " name="year">
        <br><strong>The date of birth entered is in the
future!</strong>
        <br><input type="submit" value="Search for
films">
        <input type="reset" value="Clear Form">
        </p>
    </form>
</body>
</html><?
} else {
    // do something with the data
}

```

d) i)

```

$link = mysql_connect('localhost:3307', 'user',
'password');

```

(Note: mysql_connect and server name localhost or 127.0.0.1 are required)

d) ii)

```

// assume connection already established from part d) i)
$result = mysql_select_db("Films", $link);
if (! $result) {
    echo "Failed to connect to database.\n";
}
else
{
    $result = mysql_query($query, $link);
    if (! $result) {
        echo "<p>MySQL Error: " . mysql_error($link) .
"</p>\n";
    }
    else
    {
        $all_results = mysql_fetch_array($result);
        if(empty($all_results))
        {
            echo="There are no films listed.\n";
        }
        else
        {
            echo "<table>\n";
            echo " <tr><th>Title</th><th>Rating</th></tr>\n";
            foreach($all_results as $filename, $rating)
            {
                echo "
<tr><td>$filename</td><td>$rating</td></tr>\n";
            }
        }
    }
}

```



```

    }
    echo "</table>\n";
  }
}
mysql_close($link);

```

d) iii)

```

// assume connection already established from part d) i)
$dateOfBirth = $year . "-" . $month . "-" . $day;
$ageInSeconds = strtotime("now") - strtotime($dateOfBirth);
$ageInYears = (((($ageInSeconds / 60) / 60) / 24) / 365.2425);
$okToProceed = true;

if($ageInYears>=18) {
  $query="SELECT * from FilmRatings;"
} else if ($ageInYears>=15) {
  $query="SELECT * from FilmRatings where rating='UC' or
rating='PG' or rating='12' or Rating='15';";
} else if ($ageInYears>=13) {
  $query="SELECT * from FilmRatings where rating='UC' or
rating='PG' or Rating='12';";
} else if ($ageInYears>=12) {
  $query="SELECT * from FilmRatings where rating='UC' or
rating='12';";
}
  else if ($ageInYears>=5) {
$query="SELECT * from FilmRatings where rating='UC';";
} else {
  echo="This person is not old enough to watch any films.\n";
  $okToProceed=false;
}

if($okToProceed) {
$result = mysql_select_db("Films", $link);
if (! $result) {
  echo "Failed to connect to database.\n";
}
else
{
  $result = mysql_query($query, $link);
  if (! $result) {
    echo "<p>MySQL Error: " . mysql_error($link) . "</p>\n";
  }
  else
  {
    $all_results = mysql_fetch_array($result);
    if(empty($all_results))
    {
      echo="This person is not old enough to watch any
films.\n";

```

```

}
else
{
    echo "<table>\n";
    echo "  <tr><th>Title</th><th>Rating</th></tr>\n";
    foreach($all_results as $filename, $rating)
    {
        echo "  <tr><td>$filename</td><td>$rating</td></tr>\n";
    }
    echo "</table>\n";
}
}
mysql_close($link);
}

```

A2. Examiners' Guidance Notes

This question was attempted just over half of the candidates, with a pass rate of nearly 60%.

The evidence shows that candidates gave good answers for parts a, b and c (i), but struggled in the other parts. Sub-question (d (i)) was relatively popular in term of the right answer.

Most candidates are more confident with HTML than a programming language which suggests that candidates would benefit from revising for coding-oriented questions.

A3. Two concerns for web services that attract high volumes of visitors are *availability* and *performance*. ISO27000:2016 defines availability as the “*property of being accessible and usable upon demand by an authorized entity*”

a) Individual servers may use *RAID storage technology* to improve their availability and/or performance.

i) With the aid of a diagram, briefly describe the following three standard RAID categories:

- **RAID 0**
- **RAID 1**
- **RAID 5**

(3 marks)

ii) Of the three categories of RAID listed in part a) above, **state** and **justify** with reasons which offers the best improvement from a **performance** perspective.

(2 marks)

iii) Of the three categories of RAID listed in part a) above, **state** and **justify** with reasons which offers the best improvement from an **availability** perspective.

(2 marks)

iv) Considering your answers to part ii) and iii), and also value for money, **make a recommendation** which scheme should be adopted for a web site which stores and streams user-created videos.

(1 marks)

b) Another approach to availability is to have multiple servers in place with load balancing.

i) Explain what is meant by **load balancing**, and how this can improve website availability.

(2 marks)

ii) When a company has only two servers, it may choose to keep the other server entirely unused to act as a “hot swap” instead of load balancing between the two. Provide TWO benefits and TWO drawbacks of this approach.

(4 marks)

iii) With the aid of an example, **briefly explain** how cloud-based solutions (e.g. Amazon Web Services) can be used to improve service availability, compared to building a resilient system in-house.

(3 marks)

c) A web service provider has had reports from users that their service is “running slow” from their perspective during the middle of the day.

i) Provide TWO hypotheses of why this might be happening.

(2 marks)

ii) Propose a way to test each hypothesis.

(4 marks)

iii) Recommend one solution for each hypothesis, if it is validated.

(2 marks)

Answer Pointers

a) i) RAID 0 – striping without mirroring or parity. Improved throughput, but no redundancy.

RAID 1 – mirroring, i.e. data written identically to both drives. Full redundancy against a single drive failure.

RAID 5 – block level striping with distributed parity. Requires at least three disks, redundancy against a single drive failure.

[0.5 marks for each description, 0.5 for each suitable diagram]

a) ii) RAID 0 offers the best performance, as it can read from both drives to access data, increasing throughput [1 mark for correct identification, 1 mark for explanation]

a) iii) RAID 1 offers the best availability, as it can suffer a single disk failure with no decrease in performance (unlike RAID 5 which runs at reduced performance until rebuilt with a new drive) [1 mark for correct identification, 1 mark for explanation]

a) iv) RAID 1 is more expensive, since it requires 2n drives to provide n storage. RAID 0 offers no redundancy, so a disk failure would lose data. The best compromise is RAID 5 which requires 3n drives to provide 2n in storage and has some improved performance in reading data. [1 mark for stating solution with reason, 0.5 marks if not justified].

b) i) Load balancing is the distribution of work across multiple resources (network infrastructure, computers, CPUs, disks) to optimise resource use and throughput, and minimise response time. It can also provide redundancy in the case that one of the balanced resources fails [1 mark for definition, 1 mark for benefits to availability].

b) ii)

Benefits	Drawback
Secondary system is immediately available to take over in case of failure of the primary	Secondary system may not have been proven reliable under “real” system loads and may have issues previously undetected.
System is unused, so has not suffered “wear and tear” that it might have if it were in active use.	Resource use for the service (and risk of failure of a component) may be greater as only one server is taking the load as opposed to distributing it.

b) iii) Three marks available for three valid observation or benefits raised. Typically, this will describe a characteristic of a suitable cloud-based provider, outline the benefit and then show how it is more effective/cheaper per service than an in house system built to the same specification.

One example answer considering AWS. AWS resources are organised in regions that supply a similar set of services, and is comprised of two or more availability zones, each of which having one or more servers. [1 mark for detail]. If you configure your service to have resources spread across multiple availability zones, it reduces the chance that any network outage will affect the whole service. [1 mark for benefit]. With an in-house system, you would have to build this redundancy yourself no matter how much it is used, whereas AWS can build it once for all customers, giving economies of scale [1 mark].

c) i) Any two reasonable hypotheses are acceptable, e.g.:

- large spike of users accessing the service on their lunch break

- routine maintenance e.g. disk defragmenting occurring on the servers at the time of the slowness
- hardware is reaching end of life and suffering degraded performance
- users PCs have maintenance tasks running in the background
- malware present on the user's PC
- Problem with the routers on either user/server side

[1 mark for each hypothesis, hypotheses which are too similar will only get credit for the best one]

c) ii) 2 marks for each hypothesis considered, to a maximum of 4 marks. This will depend on the hypothesis provided but should give a clear way to test the hypothesis [1 mark] and some reasonable attempt to rule out other factors or to find additional supporting evidence [1 mark]. Solutions which are too similar will only get credit for the best solution.

Example: To see if the large spike of users on lunch is the factor, take performance measures to indicate numbers of users in the morning, at lunchtime and afternoon, and plot against a graph over time. Perhaps compare these with network utilisation and plot on the same time to show there is correlation between the two. If there is a spike at lunchtime, this is some evidence to support the hypothesis.

c) iii) The solution provided for each should be reasonable and proportional to the issue identified. Proposing full server mirroring/load balancing/etc. to compensate for a lunchtime spike is likely to be excessive, unless well justified. Solutions which are too similar will only get credit for the best solution. [1 mark for each, maximum 2 marks total]

A3. Examiners' Guidance Notes

This question was attempted by approximately one third of candidates, with a high pass rate. There is evidence that the sub-question (a.i) was well answered by most candidates. However, some did not use diagrams. Parts (a.ii), (a.iii), (a.iv) were not correctly answered.

Question (b.i) was reasonably well answered.

Section B

Answer Section B questions in Answer Book B

B4. a) An DTD element may be accompanied by a symbol such as +, * or ?. Describe the meaning of EACH of these symbols using suitable examples.

(4 marks)

- b) An XML author must decide when to use attributes to contain data and when to use nested elements. Give an example to demonstrate this and state FOUR reasons why attributes are of limited use.
- (4 marks)**
- c) The XML document in figure 4.2 contains precisely **FOUR** errors when validated against the DTD in figure 4.1. 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)**
- d) Briefly explain the benefits and usage of the empty element in an XML DTD document citing a suitable example.
- (3 marks)**
- e) Write an external DTD for the XML file in figure 4.3:
- Sequence of elements is as shown in the XML document (i.e. `cbcatalogue` is a container of `subject` elements).
 - `subject` element must be present zero or more times.
 - Each `subject` element contains a `subject_title` and zero or more `books`.
 - The two attributes in `book` are mandatory.
 - A `cover_image` for each `book` that captures attributes of height, width and location of image
 - `book_url` is optional.
 - `book_url` must have a `page` attribute.
 - For each `book`, exactly one `book_title` and one `Publisher` element must be present
 - For each `book`, there must be one or more `authors`.
 - An `author` has a `Name` and optionally a short `biography` and an `email`.
 - A `Name` consists of both a `First_name` and a `Last_name`.
 - For each `book`, there must be one or more `reviews`.
- (10 marks)**

Answer pointers

a) 1 or more, 0 or more, 0 or 1 only Inline Definition:

b)

attributes cannot contain multiple values (child elements can)
attributes are not easily expandable (for future changes)
attributes cannot describe structures (child elements can)
attributes are more difficult to manipulate by program code
attribute values are not easy to test against a DTD
using attributes as containers for data, leads to documents that are difficult to read and maintain.
use elements to describe data.
use attributes only to provide information that is not relevant to the data.

c)

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE RecipeCatalogue SYSTEM "recipe.dtd">
3 <RecipeCatalogue>
4     <recipe>
5         <recipeName>CTM</recipeName>
6         <ingredient >
7             <quantity amount= "1" unit= "kg"/>
8             <name>Chicken</name>
9         </ingredient>
10        <ingredient>
11            <quantity amount="2" unit="kg"/>
12            <name>Onion</name>
13    </recipe>
14 </RecipeCatalogue>
```

d) empty element cannot have information but can hold attributes which are used for

images,
etc

e)

```
<?xml version = "1.0" encoding="UTF-8"?>
<!ELEMENT cbcatalogue (subject*)>
<!ELEMENT subject (subject_title,book*)>
<!ELEMENT subject_title (#PCDATA)>
<!ELEMENT book (book_title,image*, author+,book_url?,publisher, review+)>
<!ATTLIST book isbn CDATA #REQUIRED edition CDATA #REQUIRED>
<!ELEMENT book_title (#PCDATA)>
<!ELEMENT image EMPTY>
<!ATTLIST image height CDATA #REQUIRED>
<!ATTLIST image width CDATA #REQUIRED>
<!ATTLIST image
src CDATA #REQUIRED >
<!ELEMENT author (name, email, bio)>
<!ELEMENT name (Last_name,First_name)>
<!ELEMENT Last_name (#PCDATA)>
<!ELEMENT First_name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT bio (#PCDATA)>
<!ELEMENT book_url EMPTY>
```

B4. Examiners' Guidance Notes

This was a popular question, attempted by almost all candidates with a high pass rate

The evidence shows that candidates demonstrated a sound knowledge of the basics of XML constructs and DTDs. There were however some candidates who were unable to reply with answers enforcing required attributes and the correct use of empty elements in writing a DTD.

- B5. a)** Briefly explain the purpose of an XSLT document in relation to an XML document. **(2 marks)**
- b)** Using the two files listed in Figures 5.1 and 5.2, `football.xml` and `football.xsl`, explain what would be seen in a browser window when the `football.xml` file is loaded. You are only required to show the content generated and not the browser components.
- (Notes: `football.xml` currently captures a player's name, previous clubs, number of red cards received to date, number of yellow cards received to date, `red_cost` is the cost of a red card and similarly for the `yellow_cost`.) **(5 marks)**
- c)** A new column, heading "Medals" is required to show the number of medals won and a corresponding message. Modify the code in `football.xsl` file to achieve this. Using the line numbers to write the new code and clearly indicate where it should be placed. The message should be "Number of medals" followed by the actual number of medals won. **(3 marks)**
- d)** Modify the code developed for part **c)** to display a message based on the number of medals won followed by the actual number of medals won. The message must be one of the following:
- 0 medals: "No medals as yet, there is time";
 - 1 medal: "You are a WINNER"
 - 2 to 4 medals: "You are a SUPER WINNER";
 - 5 or more medals: "You are a Champion of Champions"
- (5 marks)**
- e)** Create another new column with the heading "Disciplinary Record and Fines" in which the following information is to be shown:
- number of red cards received;
 - total cost of sending offs;
 - number of yellow cards received;
 - total cost of warnings;
 - grand total of all penalties.

(Note: A red card means the player is sent off and will incur a penalty as indicated in the xml file, a yellow card means that the player receives a warning which too has a penalty)

(5 marks)

- f) Modify the name column to show player photograph and make the image an active link to player's Twitter account.

(5 marks)

Answer Pointers

a)

XSLT, XML – XSLT allows translation of XML docs into HTML

b)

MyClub Player Data 2018

Name	Previous Clubs
Peter Pan	1. Southend United 2. Exeter City 3. Colchester United
David Bond	1. Colchester United 2. Canterbury Town 3. Bristol City
James Kraft	1. Bristol City 2. Swansea City
Adam Chance	1. Notts Conty 2. Grimsby Town

c - f)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
```

```
  <xsl:template match="/">
```

```
    <html>
```

```
      <head>
```

```
        <title>FootyNew</title>
```

```
        <!--<style>
```

```

        th { padding: 20px; font-family: Arial, sans-
        serif;color:green}

        h2 { color: red}
        .altd {color:green}
        .gold {color:gold}

    </style>-->
        <link rel="stylesheet" type="text/css"
href="footy.css"/>
    </head>
    <body>
        <h2>MyClub Player Data 2018</h2>
        <table border="5">
            <tr bgcolor="yellow">
                <th align="center">Name</th>
                <th align="center">Previous Clubs</th>
                <th align="center">Medals</th>
                <th align="center">Disciplinary
Record</th>

            </tr>

            <xsl:for-each select="footballPlayer/player">
                <tr>

                    <td >
                        <div class="altd">

                            <xsl:value-of
select="name/first"/>

                            <xsl:text>&#xa0;</xsl:text>

                            <xsl:text>&#xa0;</xsl:text>

                            <xsl:value-of
select="name/last"/>

                            <br/>

                            <xsl:element name="a">

                                <xsl:attribute
name="href">

                                    <xsl:text>https://twitter.com/</xsl:text>

                                <xsl:value-
of select="twitter"/>

                                </xsl:attribute>

                                <img>

                                    <xsl:attribute name="src">

                                        <xsl:value-of select="image/@src"/>

```

```

</xsl:attribute>

<xsl:attribute name="width">
<xsl:value-of select="image/@width"/>
</xsl:attribute>

<xsl:attribute name="height">
<xsl:value-of select="image/@height"/>
</xsl:attribute>
</img>
</xsl:element>
</div>
</td>
<TD>
<ol>
<xsl:for-each
select="previous_clubs/club">
<li>
<xsl:value-
of select="."/>
</li>
</xsl:for-each>
</ol>
</TD>
<TD>
<xsl:choose>
<xsl:when test="medals
>4">
<div
class="gold">You are a
<em>Champion of Champions</em>
</div>
with
<xsl:value-of
select="medals"/>
medals
</xsl:when>
<xsl:when test="medals
>3">
You are a
<em>
WINNER</em>
with

```

```

select="medals"/>
<xsl:value-of
medals
</xsl:when>
<xsl:when test="medals
>2">
You are a
<em>
WINNER</em>
with
<xsl:value-of
medals
</xsl:when>
<xsl:when test="medals
>1">
You are a
<em>
WINNER</em>
with
<xsl:value-of
medals
</xsl:when>
<xsl:when test="medals
>0">
You are a
<em>
WINNER</em>
with
<xsl:value-of
medal
</xsl:when>
<xsl:otherwise>No
Medals, There is still time.</xsl:otherwise>
</xsl:choose>
</TD>
<pre>
<td>
no. of red cards <xsl:value-of
cost of sending off is: &#xa3;
<xsl:value-of
select="cards/red_card/@red_cost*cards/red_card"/><br/>
no. warnings <xsl:value-of
cost of warnings is: &#xa3;
<xsl:value-of
select="cards/yellow_card/@yellow_cost*cards/yellow_card"/><br/>





```

```

Total Cost of indiscipline is
&#xa3; :<xsl:value-of
select="cards/yellow_card/@yellow_cost*cards/yellow_card+cards/red_card/@red_
cost*cards/red_card"/>
<br/>
</td>
</pre>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

MyClub Player Data 2018

Name1	Previous Clubs	Medals	Disciplinary Record
Peter Pan 	1. Southend United 2. Exeter City 3. Colchester United	You are a <i>WINNER</i> with 1 medal	no. of red cards 1 cost of sending off is: £ 500 no. warnings 5 cost of warnings is: £ 500 Total Cost of indiscipline is £ :1000
David Bond 	1. Colchester United 2. Canterbury Town 3. Bristol City	No Medals, There is still time.	no. of red cards 0 cost of sending off is: £ 0 no. warnings 10 cost of warnings is: £ 1000 Total Cost of indiscipline is £ :1000
James Kraft 	1. Bristol City 2. Swansea City	You are a <i>Champion of Champions</i> with 8 medals	no. of red cards 4 cost of sending off is: £ 2000 no. warnings 0 cost of warnings is: £ 0 Total Cost of indiscipline is £ :2000
Adam Chance 	1. Notts Conty 2. Grimsby Town	You are a <i>WINNER</i> with 1 medal	no. of red cards 0 cost of sending off is: £ 0 no. warnings 0 cost of warnings is: £ 0 Total Cost of indiscipline is £ :0

B5. Examiners' Guidance Notes

This question was attempted by less than half of candidates with a good pass rate.

To answer this question, candidates need to be competent in writing code. The question is about rendering XML data to output HTML pages with images, hyperlinks, calculated columns and comments. There is evidence that most candidates correctly rendered the basic page but struggled with adding details which required the use of for loops, nested for loops and multi conditional if statement.

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 6 Professional Graduate Diploma in IT

FIGURES TO ACCOMPANY 2018 EXAMINATION PAPER IN
WEB ENGINEERING

Film Rating Search

form.html

Search Films

Enter the film title:

Enter your date of birth:

Figure A2.1 – Sample HTML form for question A2 a)

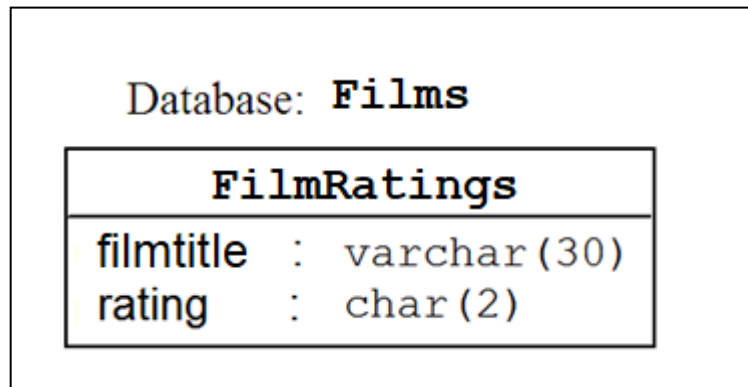


Figure A2.2 – Database Entity-Attribute diagram for question A2 d)

Rating	UC	12	PG	15	18
Minimum age	5 years	12 years	13 years	15 years	18 years

Figure A2.3 Age rating categories and minimum age required to watch the film

```

<?xml version = "1.0" encoding="UTF-8"?>
<!ELEMENT RecipeCatalogue (recipe+)>
<!ELEMENT recipe (recipeName, ingredient+)>
<!ELEMENT recipeName (#PCDATA)>
<!ELEMENT ingredient (quantity*,name)>
<!ELEMENT quantity (#PCDATA)>
<!ATTLIST quantity amount CDATA #REQUIRED unit CDATA "g">
<!ELEMENT unit (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT name (#PCDATA)>

```

Figure 4.1: XML simple DTD

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE RecipeCatalogue SYSTEM "recipe.dtd">
3 <RecipeCatalogue>
4   <recipe>
5     <recipeName>CTM</recipeName>
6     <ingredient>
7       <quantity amount = 1 unit="kg" />
8       <Name>Chicken</Name>
9     </ingredient>
10    <ingredient>
11      <quantity amount ="2" unit="kg" />
12      <name>Onion</name>
13    </ingredients>
14  </recipe>
15 </Recipecatalogue>

```

Figure 4.2: XML document with errors


```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE cbcatalogue SYSTEM "cbcatalogue.dtd">
<?xml-stylesheet type="text/xsl" href="cbbookxx.xsl"?>
<cbcatalogue>
  <subject>
    <field>Web Technology </field>
    <book isbn="0137001312" edition="1st">
      <title>JavaScript for Programmers</title>
      <image src="xml.jpg" height="100" width="100" />
      <author>
        <name>
          <Last_name>Deitel</Last_name>
          <First_name>Paul</First_name>
        </name>
        <email>deitel@deitel.com</email>
        <bio>Paul Deitel, Chief Technical Officer of Deitel and Associates.</bio>
      </author>
      <author>
        <name>
          <Last_name>Deitel</Last_name>
          <First_name>Harvey</First_name>
        </name>
        <email>deitel@deitel.com</email>
        <bio>Chairman and Chief Strategy Officer of Deitel and Associates.</bio>
      </author>
      <book_url page="http://www.deitel.com/books"/>
      <book_url linkname="JavaScript for Programmers"/>
      <publisher>Prentice Hall</publisher>
      <review>Demonstrating use of commonly available tools to create dynamic webpages &
        server side technologies.</review>
    </book>
    <book isbn="067232797" edition="3rd">
      <title>Sams Teach Yourself XML in 24 Hours</title>
      <image src="javascript.jpg" height="100" width="100" />
      <author>
        <name>
          <Last_name>Morrison</Last_name>
          <First_name>Michael</First_name>
        </name>
        <email>morrisonm@michaelmorrison.com</email>
        <bio>Michael Morrison is a professional Java programmer.</bio>
      </author>
      <publisher>SAMS</publisher>
      <review>As an introductory text on XML, it delivers.</review>
    </book>
  </subject>
  <subject>
    <field>Databases</field>
    <book isbn="9781408007686" edition="2nd">
      <title>Database Management Systems</title>
      <image src="dbms.jpeg" height="100" width="100" />
      <author>
        <name>
          <Last_name>Ward</Last_name>
          <First_name>Patricia</First_name>
        </name>
        <email>emea.info@cengage.com</email>
      </author>
      <book_url page="http://www.cengage.co.uk/fasttrack"/>
      <book_url linkname="Database Management Systems"/>
      <publisher>Course Technology</publisher>
      <review>Exactly what is required in an introductory database course.</review>
    </book>
  </subject>
</cbcatalogue>

```

Figure 4.3: XML cbcatalogue

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="football.xsl" ?>

<footballPlayer>
  <player>
    <name>
      <first>Peter</first>
      <last>Pan</last>
    </name>
    <image src="p1.png" height="100" width="100"/>
    <twitter>@peterPan</twitter>
    <previous_clubs>
      <club>Southend United</club>
      <club>Exeter City</club>
      <club>Colchester United</club>
    </previous_clubs>
    <medals>1</medals>
    <cards>
      <red_card red_cost="500">1</red_card>
      <yellow_card yellow_cost="100">5</yellow_card>
    </cards>
  </player>

  <player>
    <name>
      <first>David</first>
      <last>Bond</last>
    </name>
    <image src="p2.png" height="100" width="100"/>
    <twitter>@davidBond</twitter>
    <previous_clubs>
      <club>Colchester United</club>
      <club>Canterbury Town</club>
      <club>Bristol City</club>
    </previous_clubs>
    <cards>
      <red_card red_cost="500">0</red_card>
      <yellow_card yellow_cost="100">10</yellow_card>
    </cards>
  </player>

  <player>
    <name>
      <first>James</first>
      <last>Kraft</last>
    </name>
    <image src="p3.png" height="100" width="100"/>
    <twitter>@JKraft</twitter>
    <previous_clubs>
      <club>Bristol City</club>
      <club>Swansea City</club>
    </previous_clubs>
    <medals>8</medals>
    <cards>
      <red_card red_cost="500">4</red_card>
      <yellow_card yellow_cost="100">0</yellow_card>
    </cards>
  </player>

  <player>
    <name>
      <first>Adam</first>
      <last>Chance</last>
    </name>
    <image src="p4.png" height="100" width="100"/>
    <twitter>@adamChance</twitter>
    <previous_clubs>
      <club>Notts Conty</club>
      <club>Grimsby Town</club>
    </previous_clubs>
    <medals>1</medals>
    <cards>
      <red_card red_cost="500">0</red_card>
      <yellow_card yellow_cost="100">0</yellow_card>
    </cards>
  </player>
</footballPlayer>

```

Figure 5.1 football.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3
4     <xsl:template match="/">
5         <html>
6             <head/>
7             <body>
8                 <h2>MyClub Player Data 2018</h2>
9                 <table border="1">
10                    <tr bgcolor="yellow">
11                        <th align="left">Name</th>
12                        <th align="center">Previous Clubs</th>
13                    </tr>
14
15                    <xsl:for-each select="footballPlayer/player">
16                        <tr>
17                            <td>
18                                <xsl:value-of select="name/first"/>
19                                <xsl:text> </xsl:text>
20                                <xsl:value-of select="name/last"/>
21                                <br/>
22                            </td>
23                            <td>
24                                <ol>
25                                    <xsl:for-each select="previous_clubs/club">
26                                        <li>
27                                            <xsl:value-of select="."/>
28                                            <br/>
29                                            <br/>
30                                        </li>
31                                    </xsl:for-each>
32                                </ol>
33                            </td>
34                        </tr>
35                    </xsl:for-each>
36                </table>
37            </body>
38        </html>
39    </xsl:template>
40 </xsl:stylesheet>

```

Figure 5.5 football.xsl