**Birkbeck**
**(University of London)**

**MSc and MRes Examination for Internal Students**

***MSc in Advanced Information Systems***
***MSc in Intelligent Information Systems***
***MSc in Web Information Management***
***MRes in Computer Science***
***MSc in E-Business***

**School of Computer Science and Information Systems**

# Development of Internet Applications (COIY032P)

**Date of examination: Wednesday 28 May 2008**
**Duration of paper: 14:30–16:30**

*There are six questions on this paper. Candidates should attempt any FOUR of them. Calculators are not permitted.*

1. Consider the following document type declaration and DTD for an XML document representing stock held by a car dealer:

```
<!DOCTYPE stock [
<!ELEMENT stock     ((new-car | used-car)*) >
<!ELEMENT new-car   (model, price) >
<!ELEMENT used-car  (model, price, mileage, condition?) >
<!ELEMENT model     (#PCDATA) >
<!ELEMENT mileage   (#PCDATA) >
<!ELEMENT price     (#PCDATA) >
<!ELEMENT condition (#PCDATA) >
]>
```

(a) What difficulties do you forsee if the car dealer wanted instead to represent each car using a `car` element, with “new” or “used” being represented by an attribute? (4 marks)

(b) If you were told that the only allowed values for `condition` were “as new” and “good”, how would you modify the above DTD to ensure that `stock` documents adhered to this requirement? (6 marks)

(c) Given a document conforming to the above DTD and assuming no context node, give an XPath expression that will return the number of models of used cars whose mileage is less than 20000. (5 marks)

(d) Assume that we need to transform an XML document conforming to the above DTD as follows. The document (root) element is still `stock`, but we want all `new-car` elements to precede all `used-car` elements in the output. In addition, we only want in the transformed document new cars whose price is less than 10000 and used cars whose condition is “as new”. Write a single XSLT rule that will perform the transformation (do not include the `stylesheet` element). (10 marks)

2. (a) Using the XML schema definition language, write down appropriate declarations for a `book` element and a `person` element, which are to appear in the same document and are constrained as follows. A `book` element must have a `title` element and a `year` element as children. A `person` element must have a `title` element, which must contain either “Mr” or “Ms”, and a `name` element as children. Would such a definition be possible using a DTD? Explain your answer. (18 marks)

(b) In URI syntax, what are the hash symbol (`#`) and the question mark used for? (4 marks)

(c) In HTTP/1.0 the end of a resource is always implicitly indicated by the connection being closed by the server. Given that this is not necessarily the case with HTTP/1.1, how is an end of resource indicated to the client? (3 marks)

3. (a) Explain why certain tags can be omitted in HTML but not in XHTML. (4 marks)

(b) Explain how a *namespace* is declared in an XML document. (8 marks)

(c) When considering expressions in XPath, explain what is meant by the statement that the `preceding`, `following`, `ancestor`, `descendant` and `self` axes together *partition* an XML tree into 5 subtrees. (4 marks)

(d) Explain the similarities and differences between TCP and UDP. (9 marks)

4. (a) What does the acronym *MIME* stand for? What limitations was MIME designed to overcome? (9 marks)

   (b) Explain how the basic HTTP/1.1 caching algorithm works, indicating which steps correspond to a cache hit and which to a cache miss. (16 marks)

5. Consider the following user-defined Javascript function

```
function loadXMLHTTP(url) {
  if (window.XMLHttpRequest) {
    xmlhttp = new XMLHttpRequest();
  } else if (window.ActiveXObject) {
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  } else {
    alert("No browser support for XML-HTTP-request object")
  }

  xmlhttp.open("GET", 'proxy.php?url=' + escape(url), false);
  xmlhttp.send(null);
  return xmlhttp.responseXML;
}
```

along with the following fragment of PHP code stored as `proxy.php`

```
<?php
header('Content-type: text/xml; charset=utf-8');
$url = $_GET['url'];
echo file_get_contents($url);
?>
```

Describe in detail the purpose of each line of code in each of the above fragments. This should include explanations of each object, method or value used, relating them to HTTP requirements or conventions where appropriate. You should also explain why `proxy.php` is needed in order to achieve the overall functionality. (25 marks)

6. Consider the following (simplified) DTD for representing information about online auctions, where `site` is the document (root) element:

```
<!ELEMENT site            (people, open_auctions)>
<!ELEMENT people          (person*)>
<!ELEMENT person          (name)>
<!ATTLIST person          id ID #REQUIRED>
<!ELEMENT open_auctions   (open_auction*)>
<!ELEMENT open_auction    (initial, reserve?, bidder*, seller)>
<!ELEMENT bidder          (personref, increase)>
<!ELEMENT seller          EMPTY>
<!ATTLIST seller          person IDREF #REQUIRED>
<!ELEMENT personref       EMPTY>
<!ATTLIST personref       person IDREF #REQUIRED>
```

(a) Write down an XML document, valid with respect to the above DTD, representing the following information. There are two open auctions. For the first, the seller is a person named Jack; for the second auction, the seller is a person named Jill. For the first auction, the initial price is 20 and there is one bidder, namely Jill, with an increase of 10. For the second auction, the initial price is 10 and there are no bidders. (12 marks)

(b) Now compose a set of XSL stylesheet rules (do not worry about the stylesheet element itself) that, when given a document valid with respect to the above DTD, will output an HTML table. Assuming the stylesheet is used on the document described in Part 6a with an additional bid by Jack on the item he is selling (with an increase of 15), the output would look as follows:

| **Seller** | **Initial price** | **Bidder** | **Increase** |
|---|---|---|---|
| Jack | 20 | | |
| | | Jill | 10 |
| | | Jack | 15 |
| Jill | 10 | | |

In other words, the table contains a header row and one row for each open auction, containing the name of the seller and the initial price. Following the row for each auction, the table contains rows for each bidder in that auction, with the name of the bidder and the increase. (13 marks)