UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2011/2012 – 3rd Year Examination – Semester 5

IT5404: Internet Application Development

Structured Question Paper

17th March 2013

(TWO HOURS)

To be completed by the candidate

BIT Examination Index No:

Important Instructions:

- The duration of the paper is **2 (Two) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **17 pages**.
- **Answer all 4 questions: Each question carries 25 marks.**
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.

Questions Answered

Indicate by a cross (×), (e.g. $\cancel{1}$) the numbers of the questions answered.

To be completed by the candidate by marking a cross (×).	1	2	3	4	
To be completed by the examiners:					

Index No: ………………...

1) (a) Describe the Client-Server, N-Tier and P2P architectures of Web Applications.

(5 marks)

ANSWER IN THIS BOX

• Client/server architecture describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfils the request.

• In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations.

• N-Tier architecture is also called as Multi-Tier architecture, which contains a Presentation Tier, Business Tier and Persistence Tier.

(b) Discuss three major challenges faced in the process of adopting Component Based/ Service-oriented Architecture. If you are building a new application and choose to ignore the fundamental design characteristics of Component Based/ Service-oriented Architecture, state three problems you will likely face in future.

(10 marks)

ANSWER IN THIS BOX

1. **ensuring adequate staff training and maintaining the level of discipline required to ensure the services that are developed are reusable.**
2. **managing short term costs- Building an SOA is not cheap, reengineering existing systems cost money and the payback becomes larger over time.**
3. **Some applications need to be modified in order to participate in the SOA. Some applications may need additional programs for the purpose of service-enablement.**

The problems that may be faced in future when the basic design characteristics of service –oriented architecture have been ignored are

- Significant performance challenges when adding service interface layers to the application;

- A less than ideal design, since Web services eventually will be required to compensate for incompatible legacy design characteristics within the application;

- Additional cost and effort to redesign the application to improve performance and provide a better fit for a service interface layer;

- Difficulties during integration with standard business services or other exposed Web services; and

- Not accessible by other business applications, that may wish to use a standard function in the application, due to non exposure.

(c) XML is not just for Web pages. List five (5) other common uses of XML and briefly explain each.

(5 marks)

ANSWER IN THIS BOX

Information identification: **One can define his own markup so that he can define meaningful names for all the information items.**

Information storage: **Because XML is portable and non-proprietary, it can be used to store information across any platforms. Because it is backed by an international standard, it will remain accessible and processable as a data format.**

Information structure: **XML structures can nest so that they can be used to store and identify any kind of hierarchical information, especially long, deep or complex document sets or data sources, making it ideal for an information-management back-end to serve the Web. This is one if its most common Web applications, with a transformation system to serve it as HTML until such time as browsers are able to handle XML consistently.**

Publishing: **By combining the three previous answers (identity, storage, and structure) it is possible to get all the benefits of robust document management and control (with XML) and publish to the Web (as HTML) as well as to paper (as PDF) and to other formats (eg Braille, Audio, etc) from a single source document by using the appropriate style sheets.**

Messaging and data transfer: **XML is also very heavily used for enclosing or encapsulating information in order to pass it between different computing systems which would otherwise be unable to communicate because of their proprietary or secret data formats. By providing a lingua franca for data identity and structure, XML provides a common envelope for inter-process communication (messaging).**

Web services: **Building on all of these, as well as its use in browsers, machine-processable data can be exchanged between consenting systems, where prior to that it was only comprehensible by humans (HTML). Weather services, e-commerce sites, blog newsfeeds, AJaX sites and thousands of other data-exchange services use XML for data management and transmission and the web browser for display and interaction.**

(d) Describe the need for and role of middleware in Web Applications.

(5 marks)

ANSWER IN THIS BOX

Middleware contains the common logic/ functionalities required by the applications. Hence the developer need not to worry about implementing those functionalities in the business logic of the web application.

2) (a) For controlling formatting and appearance in XML we need to provide a style sheet or use XSLT. In HTML, however, we can do the same without either of these. Explain why this is so.

(5 marks)

ANSWER IN THIS BOX

In HTML, default styling was built into the browsers because the tag set of HTML was predefined and hardwired into browsers. In XML, where you can define your own tag set, browsers cannot possibly be expected to guess or know in advance what names you are going to be used and what they will mean. Therefore you need a style sheet or XSLT if you want to display formatted text.

(b) Write both a DTD and a Schema for the following XML code.

```
<Menu>
<Choice><Option>Bread</Option></ Choice >
< Choice ><Option>Butter</Option></ Choice >
< Choice ><Option>Jam</Option></ Choice >
</Menu>
```

(5 marks)

ANSWER IN THIS BOX

```
<!ELEMENT Menu (Choice)+>
<!ELEMENT Choice (Option)>
<!ELEMENT Option (#PCDATA)>
```

(c) List the relevant Predefined Entities available in XML for the following characters:

i) Less-than symbol (<)

ii) Greater-than symbol (>)

iii) Quote symbol (")

iv) Apostrophe symbol (')

v) Ampersand symbol (&)

(5 marks)

ANSWER IN THIS BOX

&lt;
&gt;
&quot;
&apos;
&amp;

(d) Explain the purpose of a CDATA section in XML. Write an example to illustrate a CDATA section.

(5 marks)

ANSWER IN THIS BOX

• CDATA means character data.

• CDATA is text that will NOT be parsed by a parser.

Tags inside the text will NOT be treated as markup and entities will not be expanded.

• Like unparsed entities, a XML processor will not process what is written inside a CDATA section.

Example of a CDATA section is as follows.

```
This is a face:
<![CDATA[
*****
* @ @ *
* ) *
* ~~~ *
*****
]]>
```

(e) Explain the correctness or otherwise of the following statement giving reasons.

"XML is Not a Replacement for HTML"

(5 marks)

ANSWER IN THIS BOX

• XML is a complement to HTML.
• It is important to understand that XML is not a replacement for HTML. In most web applications, XML is used to transport data, while HTML is used to format and display the data.
• A description of XML is "XML is a software- and hardware-independent tool for carrying information".

3) (a) Explain the use of default namespaces in XML and provide an example XML code to illustrate your answer.

(5 marks)

ANSWER IN THIS BOX

If an XML namespace declaration does not contain a prefix, the namespace is the default XML namespace and you refer to element type names in that namespace without a prefix.

• For example:

```
<!-- This uses
a default namespace instead of using a prefix. -->
<A xmlns="http://www.foo.org/">
<B>abcd</B>
</A>
```

(b) Write an XML Schema according to which the following XML document is valid.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE addressBook SYSTEM "AddressBook2.dtd">
<addressBook>
<friend firstName="nimal" lastName="gurusinghe"
age="39">
<address>
24, 2nd Lane, Col 7
</address>
<gender>male</gender>
</friend>

<friend firstName="ann" lastName="perera" age="29">
<address>123, Galle Rd, Dehiwala</address>
<gender>female</gender>
</friend>

<friend firstName="kamal" lastName="weerakkody"
age="25">
<address>Galle</address>
<gender>male</gender>
</friend>
</addressBook>
```

(10 marks)

ANSWER IN THIS BOX

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="addressBook">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="friend">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="address" type="xsd:string"/>
<xsd:element name="gender" type="xsd:string"/>
</xsd:sequence>
```

```
<xsd:attribute name="firstName" type="xsd:string"
use="required"/>
<xsd:attribute name="lastName" type="xsd:string"
use="required"/>
<xsd:attribute name="age" type="xsd:string"
use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

(c) Explain what the following PHP code does and list down the output of the PHP code. The address.xml XML file below will be used as the input XML file to the PHP code.

address.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Address>
<no>24</no>
<street>2nd Lane</ street >
<town>Dehiwala</ town >
<country>Sri Lanka</ country >
</ Address >
```

code:

```
<?php
//Explain code below
$parser=xml_parser_create();

// Explain code below
function start($parser,$element_name,$element_attrs)
{
switch($element_name)
{
case "ADDRESS":
echo "-- Address --<br />";
break;
case "NO":
echo "No: ";
break;
case "STREET":
echo "Street: ";
break;
case "TOWN":
echo "Town: ";
break;
case "COUNTRY":
echo "Country: ";
}
}


// Explain code below
function stop($parser,$element_name)
{
echo "<br />";
}


// Explain code below
function char($parser,$data)
{
echo $data;
}
```

```
// Explain code below
xml_set_element_handler($parser,"start","stop");

// Explain code below
xml_set_character_data_handler($parser,"char");

// Explain code below
$fp=fopen("address.xml","r");

// Explain code below
while ($data=fread($fp,4096))
{
xml_parse($parser,$data,feof($fp)) or
die (sprintf("XML Error: %s at line %d",
xml_error_string(xml_get_error_code($parser)),
xml_get_current_line_number($parser)));
}

// Explain code below
xml_parser_free($parser);
?>
```

(10 marks)

ANSWER IN THIS BOX

Initialize the XML parser
Function to use at the start of an element
Function to use at the end of an element
Function to use when finding character data
Specify element handler
Specify data handler
Open XML file
Read data
Free the XML parser

The output of the code above will be:

```
-- Address --
No: 24
Street: 2nd Lane
Town: Dehiwala
Country: Sri Lanka
```

4) (a) Describe the main features of Web Services, WSDL, SOAP and UDDI and their relationships to each other.

(10 marks)

ANSWER IN THIS BOX

Web Services: The W3C framework for Web services consists of a foundation built on top of three core XML specifications:

- **Web Services Description Language (WSDL)**
- **Simple Object Access Protocol (SOAP)**
- **Universal Description, Discovery, and Integration (UDDI)**

These technology standards, coupled with service-oriented design principles, form a basic XML-driven SOA. This first-generation Web services architecture allows for the creation of independent Web services capable of encapsulating isolated units of business functionality. It also has a number of limitations, which have been addressed in a second generation of specifications.

UDDI: Universal Description, Discovery, and Integration (UDDI) is a registry and repository for storing and retrieving Web services metadata. A registry can be used as an intermediate storage mechanism between the requester and the provider. The provider can publish the description (and associated policy and data schemas) to the registry, and the requester can search the registry for the metadata it requires. UDDI provides inquiry and publishing APIs, allowing applications to interface programmatically with a registry.

WSDL Web services need to be defined in a consistent manner so that they can be discovered by and interfaced with other services and applications. Used for defining messages, message exchange patterns, interfaces and endpoints.

The integration layer introduced by the Web services framework establishes a standard, universally recognized and supported programmatic interface. WSDL enables communication between these layers by providing standardized endpoint descriptions. The abstract interface definition is described by the interface, message and types constructs. This part of the WSDL document provides a mobile, platform-independent description of the Web service interface.
SOAP: SOAP has evolved into the most widely supported messaging format and protocol for use with XML Web services. Hence the SOAP acronym is frequently referred to as the Service-Oriented Architecture (or Application) Protocol, instead of the Simple Object Access Protocol. The SOAP specification establishes a standard message format that consists of an XML document capable of hosting RPC and document-centric data. This facilitates synchronous (request and response) as well as asynchronous (process-driven) data exchange models. With WSDL establishing a standard endpoint description format for applications, the document-centric message format is much more common.

(b) Discuss the advantages and disadvantages of Web services compared with other distributed computing models.

(5 marks)

ANSWER IN THIS BOX

- Web Services are pervasive, open standards for distributed computing and document exchange.
- Web services are more pervasive and extensible than earlier distributed computing technologies such as COM/DCOM, CORBA, Java RMI and other such technologies.
- Web services are supported by all major ISVs, including platform vendors (e.g., IBM, Microsoft, SAP, PeopleSoft, Oracle, Sun and BEA).

- Tool support for Web services and related technologies is thriving and growing.
- Extended Web services standard hold the promise of providing most of the elements of a Web services platform.
- EAI products relied on proprietary data formats and message buses, which hampered interoperability with other products and created vendor lock-in. XML and Web services allow the same open, standards-based technologies to be applied across all business domains, thus simplifying future interoperability.

(c) Explain how serialization can be carried out in XML. Illustrate your answer with an appropriate diagram.

(5 marks)

ANSWER IN THIS BOX

Serialization is the process of converting an in-memory data structure into a sequential representation. XML Serialization is the process of converting an in-memory data structure into an XML representation. XML Serialization can be used to serialize complex types as well as primitive types. For example, DataSets can be serialized into XML content and passed to or from a Web service.

(d) What do you understand by the phrase “SOAP encoding”?

(5 marks)

ANSWER IN THIS BOX

A SOAP message has no default encoding. Hence, in order to define data types used in the document, encodingStyle attribute is used. It can appear in any SOAP element.

Syntax:
soap:encodingStyle="URI"

XML format is used by SOAP for encoding data. It maps the high level data

- When a client makes a request to the server and when the server responds, at that time if power gone and client end crash,
--server never know that the client is not activated.

****