



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2017 – 3rd Year Examination – Semester 6

IT6505: Middleware Architecture

25th November, 2017
(TWO HOUR)

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 **19 pages**.
- **Answer all questions.** All questions **do not** carry equal 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.
- Programmable Calculators or any other data storage devices are **not** allowed.

Questions Answered

Indicate by a cross (x), (e.g.

X

) the numbers of the questions answered.

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

1)

- (a) (i). Define a distributed system in a single sentence. Give an example for a typical distributed system and clearly explain why such a system requires task distribution.

(05 marks)

ANSWER IN THIS BOX

A Distributed system designates a set of tightly coupled programs executing on one or more computers and coordinating their actions. These programs know about one another and co-operate to perform a task that none could carry out in isolation -> **2 marks**

3 marks for the example and explanation of a distributed system.

- (ii). Consider the statement: "If a thing goes wrong in a distributed system, it can be overcome by its architecture". Do you agree with this statement? Justify your answer by comparing and contrasting distributed systems with centralised systems.

(05 marks)

ANSWER IN THIS BOX

Yes -> **1 marks**

Distributed systems have autonomous components and also may be heterogeneous in most cases, have no single point of control but have multiple points of failure. Components are often multi-threaded and sometime components may be overloaded thus may not be able to service requests on-time. Distributed systems may require communication over a network that too is a possible point of failure. **However, fault tolerance mechanisms such as replication negate this making distributed systems more resilient -> 2 marks**

Centralized systems components are non-autonomous and are homogeneous. They have a single point of control and single point of failure. Resources are shared; hence, resources can be overloaded. Does not require components to communicate over networks. **Fault tolerance is minimal as it is not built into the architecture -> 2 marks**

- (iii). **Fault tolerance** is one form of reliability measure. List and describe two (02) other forms of reliability measures in distributed systems.

(04 marks)

ANSWER IN THIS BOX

Any 2 of the following points and 2 marks for each = 4 marks

- High availability - a highly available system provides uninterrupted service in spite of failures
- Consistency - consistency is the ability of a distributed computing system to coordinate related actions of multiple components, despite concurrency failures. It generally encompasses the ability of making a distributed system behave like a non-distributed system.
- Security - security is the ability of a system to protect data services and resources against unauthorized access

- (b) (i). Describe what is meant by Middleware in the context of a distributed application. List and explain two (02) challenges developers of distributed software would have to face in the case of non-existence of middleware.

(03 marks)

ANSWER IN THIS BOX

Middleware in the context of distributed applications is software that provides services beyond those provided by the operating system to enable the various components of a distributed system to communicate and manage data -> **1 mark**

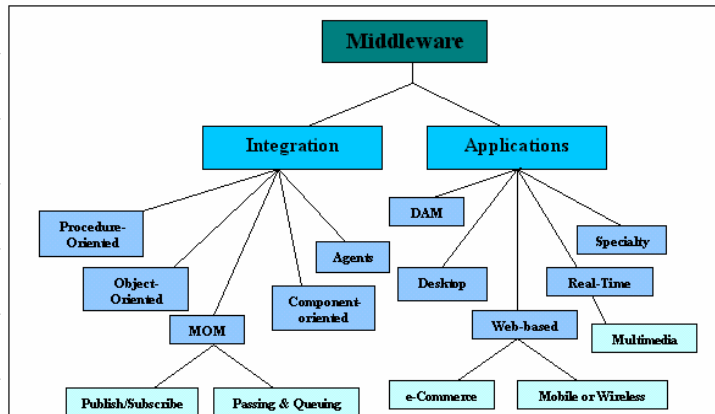
1 mark for each for two points = 2 marks

- Due to the absence of transparency having to develop components to handle distribution
- Having to develop components to have to communicate/translate between heterogeneous systems (network, data etc)
- Having to develop components to handle reliability issues of distributed networks (message delivery, error handling and recovery etc.)

- (ii). Middleware can be categorised into two major types, Integration type and Application type middleware. Give one example subcategory of middleware for each of these major categories while describing the given example middleware.

(06 marks)

ANSWER IN THIS BOX



- Integration Category: The integration type includes those middleware that have a specific way of being integrated into its heterogeneous system environment. -> **1 mark**
- Application Categories - The application classification includes middleware that fit into specific types of application functions. These middleware (Data Access, Web-based, Real- Time, Desktop, and Specialty) work specifically with an application. -> **1 mark**

2 marks for one from each category = 4

- Integration
 - Procedural: It calls procedures on remote systems and is used to perform synchronous or asynchronous interactions between applications or systems
 - MOM: is an infrastructure that supports the receiving and sending of messages over distributed applications.

- OOM: gives applications the ability to send objects and request services via an object-oriented system.
- Agent: Agents are considered a middleware that consists of several components: entities (objects, threads), media (communication between one agent and another), and laws (rules on agent's communication coordination). Media can be monitors, channels, or more complex types (e.g. pipelines).
- Component Based or Reflective Middleware: A component is described as a "program that performs a specific function and is designed in such a way to easily operate with other components and applications"
- Application
 - DAM: middleware allows for direct access to databases, providing direct interaction with them.
 - Web-based: middleware that assists the user with browsing, uses interfaces that scout ahead to find pages of interest, and discerns user's changes of interest from browsing

- (c) (i). math.x is an RPC Interface Definition Language (IDL) file. What programme should be used to compile the RPC IDL file? Furthermore indicate the output files generated by this programme and each of their main functionalities?

(02 marks)

ANSWER IN THIS BOX

rpcgen is the programme used to compile the RPC IDL file -> ½ mark

Header file math.h and used to define the function contracted -> ½ mark

Server skeleton/stub (math_svc.c) marshal and unmarshal parameters -> ½ mark

Client stub (math_clnt.c) marshal and unmarshal parameters -> ½ mark

- (ii). Consider the following code segment, which is a wrapper function that handles the calling of an RPC procedure. Clearly explain the function of each line of code in this segment.

```

1.  int add( CLIENT *clnt, int x, int y) {
2.      operands ops;
3.      int *result;
4.      ops.x = x;
5.      ops.y = y;
6.      result = add_1(&ops,clnt);
7.      if (result==NULL) {
8.          fprintf(stderr,"Trouble calling remote procedure\n");
9.          exit(0);
10.     }
11.     return(*result);
12. }.

```

(05 marks)

ANSWER IN THIS BOX

Each line of code will receive ½ mark, a total of 5 marks, excluding line 10 and 12

1. Defining the wrapper function type int, name add and parameters are the client pointer, integer values x and y
2. Operands to be passed
3. The result a pointer
4. Creating a single data structure to be sent to the server
5. – do –
6. result is equated the add_1 procedure call with the data structure for parameter passing and client
7. if the result it null
8. Print error message
9. Exit wrapper function
10. (11) return result pointer through the wrapper

2)

- (a) (i). Message brokers are a building block of message-oriented middleware. List five (05) possible actions that a message broker may perform.

(05 marks)**ANSWER IN THIS BOX****Any 5 of the following points and 1 mark for each point = 5 marks**

- Route messages to one or more destinations
- Transform messages to an alternative representation
- Perform message aggregation, decomposing messages into multiple messages and sending them to their destination, then recomposing the responses into one message to return to the user
- Interact with an external repository to augment a message or store it
- Invoke Web services to retrieve data
- Respond to events or errors
- Provide content and topic-based message routing using the publish–subscribe pattern

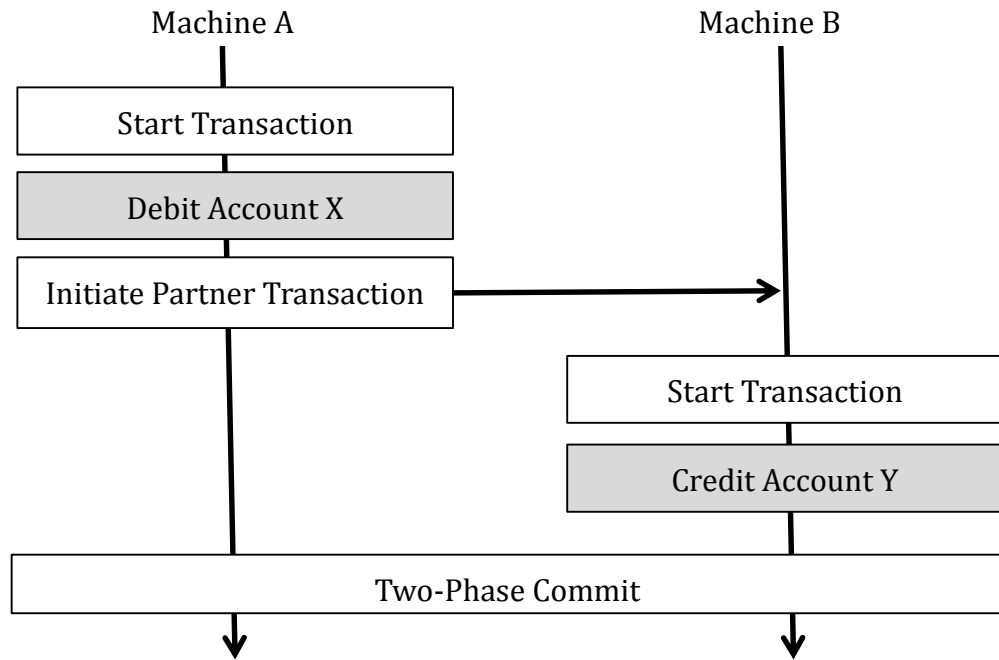
- (ii). While Message Oriented Middleware (MOM) has distinct advantages over procedural middleware of the likes of RPC, however MOM has certain disadvantages too. List one advantage and one disadvantage of MOM over RPC.

(04 marks)**ANSWER IN THIS BOX**

Unlike RPC, MOM does not have an IDL hence the developer needs to specify specific message formats due to the lack of contracts between client and server. -> **2 marks**

MOM is asynchronous and is more tolerant to failures compared to RPC -> **2 marks**

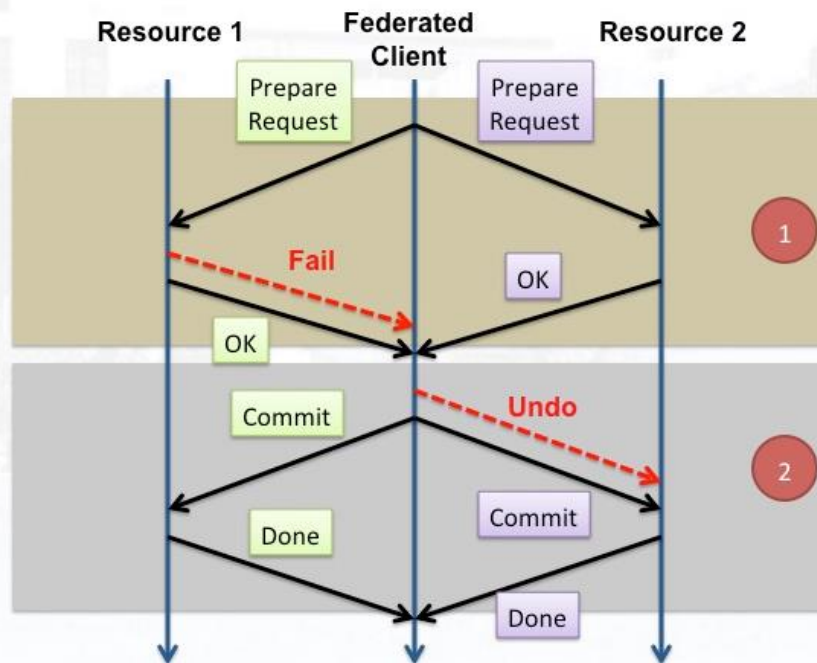
- (b) The following diagram shows how a debit/credit transaction is performed using distributed transaction processing.



- (i). Clearly explain using a diagram how the Two-Phase commit protocol works in guaranteeing a successful transaction.

(05 marks)

ANSWER IN THIS BOX

2 marks for the diagram

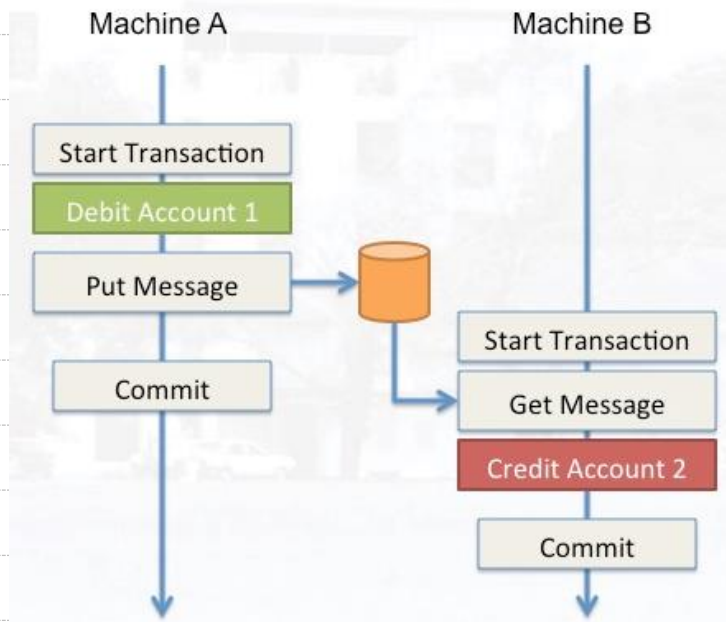
The federated client will send transaction request to PREPARE and will wait for responses from transaction processors -> **1 mark**

If all transaction processors respond OK/PREPARED then the federated client will send out the COMMIT request -> **1 mark**

Else even if one fail it would send an UNDO request to the already successful transaction processors -> **1 mark**

- (ii). Using a diagram explain how the same debit/credit transaction will be performed using Message queuing and how a successful transaction can be guaranteed in this process.

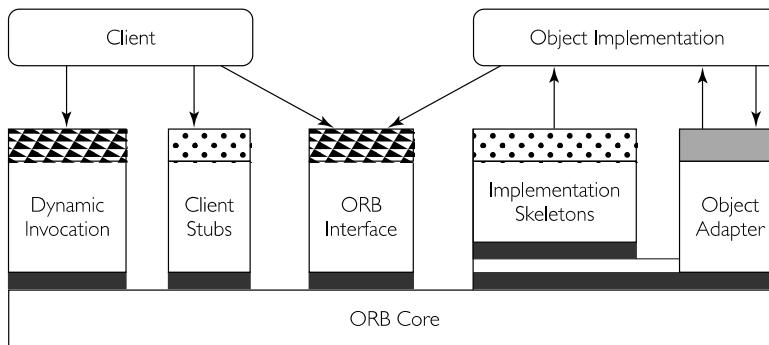
(06 marks)**ANSWER IN THIS BOX**

2 Marks for the figure

- Machine A will start the transaction and Debit Account 1 and Put a message into the queue of Machine B. -> **1 mark**
- However the message is not allowed to reach Machine B until Machine A has committed the transaction. -> **1 mark**
- Machine B will complete the transaction, however on failure it will have to send a message to Machine A to Revers the transaction -> **2 marks**

3)

- (a) (i). The following diagram depicts the CORBA Architecture. The ORB Interface, Implementation skeleton and Object Adapter are three key components of this architecture. Describe each of their functionalities within this architecture.

**(03 marks)****ANSWER IN THIS BOX**

- ORB Interface - Includes a set of types that both client and server objects use for initialization purposes. Clients can, for example, use the ORB Interface to obtain a set of initial object references; servers use the interface to select an object adapter. The ORB Interface also specifies the common root type Object whose operations and attributes every CORBA object inherits. -> **1 mark**
- Object Adapter - The PO A defines registration, activation and deactivation and supports persistent objects -> **1 mark**
- Implementation skeletons – Performs Marshalling and unmarshalling of request parameters -> **1 mark**

- (ii). Consider a student management system designed to manage student information and grades at an Educational Institute. A data structure is needed to define a student's information that includes his/her registration number, name, date of birth. The students can register for courses using the subject code and his/her student number. On completion of a course his/her grades are entered into the system. The student is also able to view his/her current Grade Point Average (GPA). Define a module, a student type, an interface, an exception handler and the required operations to carryout the actions described.

(12 marks)

ANSWER IN THIS BOX

FULL MARKS WERE GIVEN AS NO STUDENT WAS ABLE TO COMPREHEND THE QUESTION, ASSUME BECAUSE THE QUESTION DOES NOT SPECIFICALLY MENTIONING A CORBA IDL SCRIPT

```

module University -> 1 mark

    // defines a data structure date of birth
    typedef struct _DOB {

        int day;

        int month;

        int year;

    } DOB;

    // defines a data structure student
    typedef struct _Undergraduate { -> 1 mark

        string reg_no; -> 1 mark

        string name; -> 1 mark

        DOB date_of_birth; -> 1 mark

    } Undergraduate;

    // Defines the interface
    interface Academic { -> 1 mark

        readonly attribute Undergraduate Student; -> 1 mark

        attribute float gpa; (OR -> 1 mark if above attribute not
there)

        // defines an exception

        exception StudentNotFound { -> 1 mark

            string message;

        };

```

```
// defines operations

void courseRegister (in string regNo, in string courseID) raises
(StudentNotFound); -> 2 mark (1 mark for operation and 1 mark for
exception)

void courseGrade (in string regNo, in string courseID, char
grade) raises (StudentNotFound); -> 1 mark

void courseGrade (in string regNo, in string courseID, out float
gpa) raises (StudentNotFound); -> 1 mark
```

- (b) (i). Java/RMI does not require an Interface Definition Language. Explain the reason for this.

(02 marks)

ANSWER IN THIS BOX

Java was designed to be highly portable. This portability is achieved by provision of different VMs that interpret the byte code on different hardware and operating system platforms. A Java compiler creates byte- code, which is then interpreted by byte virtual machine (VM). -> 1 mark

Java includes a distinction between interfaces and classes. -> 1 mark

- (ii). Consider the code segment given below of a Java Interface which is used to instantiate remote server objects. Clearly explain function of lines 1 – 5 in this code segment.

```

1.  package cricket;
2.  import java.rmi.*;
3.  interface Team extends Remote {
4.      String name() throws RemoteException;
5.      Player[] players() throws RemoteException;
6.  }

```

(03 marks)

ANSWER IN THIS BOX

1. The scope of this is limited to the cricket package -> ½ mark
2. Makes possible that the declared interfaces are remotely accessible -> ½ mark
3. Declares the interface Team which is remote interface. The instances of the class that implement the remote interface Team would be a remote object. -> ½ mark
4. Returns the string attribute by the operation name -> ½ mark and throws RemoteException if there is an Error -> ½ mark
5. declares an array returned by the operation players -> ½ mark

4)

- (a) (i). The following diagram depicts a typical pattern in a Service Oriented Architecture. A,B and C are the key artefacts in this pattern. What do A, B and C stand for in this diagram?

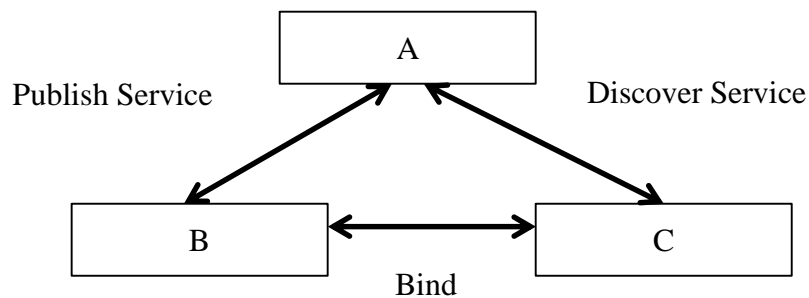


Figure 4.1

(03 marks)

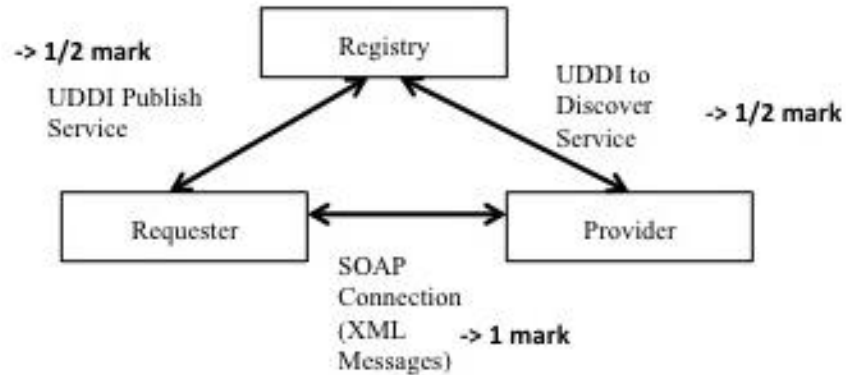
ANSWER IN THIS BOX

- A – Service Broker/Registry/Directory -> 1 mark
- B – Service Provider -> 1 mark
- C – Service Requester/Consumer -> 1 mark

- (ii). Explain how UDDI, WSDL, XML and SOAP envelopes can contribute to creating a SOAP Web service. Explain the functionality of each by elaborating the Figure 4.1 in part (i).

(06 marks)

ANSWER IN THIS BOX



- UDDI is the specification for finding and publishing web services -> 1 mark
- WSDL provides the service definition -> 1 mark
- SOAP Envelope encompasses the header and body of the SOAP message -> 1 mark
- XML is used in ASDL and SOAP messages -> 1 mark

- (iii). A web service is required to provide the information of the train schedule when provided with the departing and destination stations. The response includes the train number, departing time and arrival time at destination. If an application sends a request with *Colombo* as the departing station and *Kandy* as the destination station, create the SOAP message that the requester would receive from the service provider. (Assume that there is only one train per day and namespace may be decided by you).

(06 marks)

ANSWER IN THIS BOX

```

<? Xml version = "1.0"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2001/09/"
  soap-envelope"> -> 1 mark
  <soapenv:Header/> -> 1 mark
  <soapenv:Body> -> 1 mark
    <ns:response
      xmlns:ns="https://www.railway.gov.lk/shedule"> -> 1 mark
      <ns:train> -> ½ mark
        <ns:trainid>12345</ns:trainid> -> ½ mark
        <ns:departure>700</ns:departure> -> ½ mark
        <ns:arrival>930</ns:arrival> -> ½ mark
      </ns:train>
    </ns:response>
  </soapenv:Body>
</soapenv:Envelope>

```

- (b) (i). HATEOAS is a fundamental principle of RESTful services pattern. Clearly explain what HATEOAS is and give a code example of the use of HATEOAS in REST.

(02 marks)

ANSWER IN THIS BOX

Hypermedia as The Engine Of Application State -> ½ mark

Hypermedia is a document-centric which also provides the ability to embed links to other services and information within that document format -> ½ mark

1 mark for the tag and data with a URI

```
<order id= "111">
```

```
<customer>http://rest.example.com/customer/12345</customer>
```


- (ii). In RESTful services, both HTTP methods PUT and POST may be employed to create new resources. Even though the use of PUT method is less complex to implement, the POST method has a distinct advantage. Clearly explain what this distinct advantage is.

(02 marks)

ANSWER IN THIS BOX

When creating resources using the PUT method requires that the client has to provide the unique ID that identifies the object. -> **1 mark**

However, most application designers prefer the server to create these unique ID based on its database. POST method supports this. -> **1 mark**

- (c) An e-banking web portal allows users to pay their phone bills online. When a bill payment is made, the bank uses the phone company's RESTful service to create a payment object on the phone company's server. The payment object requires a unique payment ID string generated by the phone company's server, the payment Phone Number, Bank ID, Bank Transaction ID, Customer Name, Date & time, and Payment Amount. The RESTful service uses XML for data transfer.

- (i). Design a data format for this transaction.

(05 marks)

ANSWER IN THIS BOX

<transaction> -> **1 mark**

<phonenumber>07XXXXXXXX</phonenumber> -> **½ mark**

<bank_id="086"> -> **½ mark**

<link rel="self"

href="http://example.com/bank/086"/> -> **½ mark**

<name>XYZ Bank</name> -> **½ mark**

</bank>

<btransaction>12345678</btransaction> -> **½ mark**

<customername>John Plumber</customername> -> **½ mark**

<date>November 25, 2017 09:46</date> -> **½ mark**

<amount>LKR 2345.67</amount> -> **½ mark**

</transaction>

- (ii). If the phone company's server creates the resource, what would be the HTTP response message the client would receive from the server?

(04 marks)

ANSWER IN THIS BOX

HTTP/1.1 201 Created -> 2 mark

Content-Type: application/xml -> ½ mark

Location: http://example.com/transaction/07XXXXXXXX12345 -> ½ mark

<transaction id=07XXXXXXXX12345> -> ½ mark

<link rel="self"

href="http://example.com/transaction/07XXXXXXXX12345"/> -> ½ mark

<phonenumber>07XXXXXXXX</phonenumber>

<bank id="086">

<link rel="self"

href="http://example.com/bank/086"/>

<name>XYZ Bank</name>

</bank>

<btransaction>12345678</btransaction>

<customername>John Plumber</customername>

<date>November 25, 2017 09:46</date>

<amount>LKR 2345.67</amount>

</transaction>

(iii). Briefly explain what the HTTP respond codes 200 and 204 signify?

(02 marks)

ANSWER IN THIS BOX

200 – “OK” or the request was successfully served, response accompanied with a message

body -> **1 mark**

204 – “No content” or the request was successfully served, but no message body in

response -> **1 mark**
