



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2020 – 3rd Year Examination – Semester 5

IT6505: Middleware Architecture
Structured Question Paper
(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**.
- **Answer all questions.** All questions 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.
- Calculators are **not** allowed.
- *All Rights Reserved.*

Questions Answered

Indicate by a cross (×), (e.g.

×

) the numbers of the questions answered.

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

1) (a)

What function does the **stub** perform in RPC programming?

(4 marks)

ANSWER IN THIS BOX

The complexity of communication is completely hidden from the application by stubs that mimic the interface of the procedure calls.

→ 4 marks

(b) State *one* (01) **advantage** and *one* (01) **disadvantage** of a distributed system with compared to a centralized system for information processing.

(4 marks)

ANSWER IN THIS BOX

There can be many correct answers. Some are listed below.

Advantage:

- Distributed systems allow the sharing of information and resources, and may be composed of small, cost-effective computers that combine their processing power.
- Large computational problems can be solved at low cost by temporarily using the combined power and memory of many computers.

→ 2 marks

Disadvantage:

- Distributed systems are harder to manage because we have to address issues such as independent failures, unreliable communication, and insecure communication.
- Since more things can go wrong in distributed systems, they are intrinsically less reliable than centralized systems.
- Many mechanisms have to be built in, e.g., to handle the fact that the client and the server may fail independently (partial failures).
- The network necessary for remote communication between distributed components is another source of unreliability that distributed applications have to deal with.

→ 2 marks

(c) Distinguish between **Syntactic Transparency** and **Semantic Transparency** with respect to Remote Procedure Calls (RPC).

(6 marks)

ANSWER IN THIS BOX

- Syntactic transparency: A remote procedure and a local procedure call should have the same syntax.

→ 3 marks

- Semantic transparency: The semantics of a remote procedure call and local procedure call is identical.

→ 3 marks

- (d) What is meant by **spaghetti Integration** (Star Integration)?

(4 marks)

ANSWER IN THIS BOX

Spaghetti integration, is a process of systems integration where each system is interconnected to each of the remaining subsystems. When observed from the perspective of the subsystem which is being integrated, the connections are reminiscent of a star, but when the overall diagram of the system is presented, the connections look like spaghetti, hence the name of this method.

→ 4 marks

- (e) Consider a distributed system consisting of 5 identical servers. Each of the servers is available at any given instant with a probability of 0.9.

- i.) If the system is designed in such a way that it can be operational if **any one of the five servers** is operational, what is the **overall system availability**?

(4 marks)

ANSWER IN THIS BOX

The probability of one server being down is 0.1

→ 1 mark

The probability of all 4 servers being down simultaneously is $0.1^5 = 0.00001$

→ 2 marks

So, the probability of at least one server being available is $1 - 0.00001 = 0.99999$

→ 1 mark

- ii.) Alternatively, if the system is designed in such a way that **all five servers have to be available** for the entire system to be available, what is the **overall system availability**?

(3 marks)

ANSWER IN THIS BOX

On the other hand, the probability of all five servers being available at the same time is $0.9^5 = 0.59049$

→ 3 marks

2 (a) State *four* (04) actions performed by a **message broker** middleware.

(6 marks)

ANSWER IN THIS BOX

There can be many correct answers. Some are listed below.

- Route messages to one or more destinations
- Transform messages to an alternative representation
- Invoke Web services to retrieve data
- Respond to events or errors
- 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
- Provide content and topic-based message routing using the publish–subscribe pattern

→ 1.5x4 marks

(b) Briefly explain the main difference between **Message Queuing** and **RPC** middleware.

(4 marks)

ANSWER IN THIS BOX

Message queue can be seen as a mailbox since you can put a message in the box without the recipient being active.

→ 2 marks

In contrast, RPC needs both ends to be active at the same time, which is more like a telephone conversation.

→ 2 marks

(c) **HATEOAS** is a component of the **REST** application architecture that distinguishes it from other network application architectures. Briefly explain what is meant by HATEOAS?

(4 marks)

ANSWER IN THIS BOX

Hypermedia as The Engine Of Application State

→ 2 marks

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

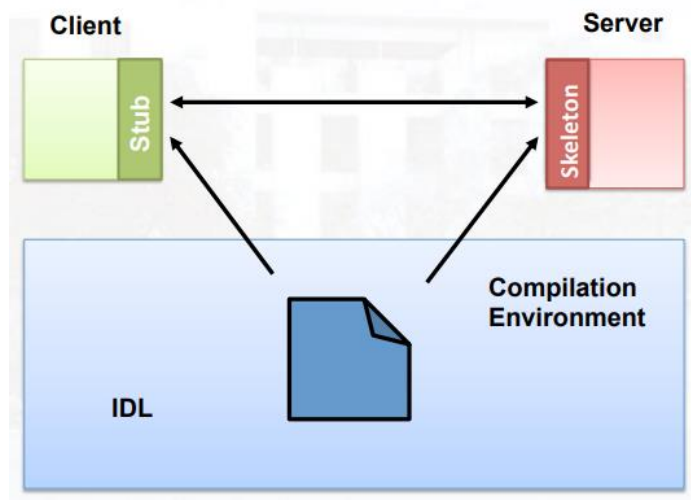
→ 2 marks

- (d) Explain briefly with the help of a diagram, the role of **Interface Definition Language (IDL)** in the design of distributed software.

(5 marks)

ANSWER IN THIS BOX

IDL is a language that is used to define the interface between a client and server process in a distributed system. Different compilers available for compiling the file and generating respective stubs on both client and server side.



→ 5 marks

- (e) “Java RMI only supports **static remote method invocations**”. Explain.

(3 marks)

ANSWER IN THIS BOX

which means that a Java remote method invocation can only be defined at the time the client object is compiled.

→ 3 marks

- (f) “Java requests have **at-most-once semantics**”. Explain.

(3 marks)

ANSWER IN THIS BOX

This means that if an error is indicated to the client, the remote method might not have been executed. If no error is indicated, clients can assume that the method has been successfully executed by the remote object. It provides no absolute message delivery guarantees since each message is delivered once (best case scenario) or not at all.

➔ 3 marks

- 3) (a)

Briefly explain the concept of in-Process servers in DCOM. State *two (02)* **practical benefits** of using **in-process servers (DLLs)** in DCOM.

(6 marks)

ANSWER IN THIS BOX

In-process servers (Dynamic Link Libraries or DLLs) can be dynamically load into the program when they are required.

□ 3 marks

There can be many practical usages. Some are listed below.

Practical Usages:

- No additional requirement of memory at runtime or disk storage as a common binary file will be there,
- All the application or clients will be using one single version of the common code provided by the DLL,
- Operating system loads only one instance of the DLL when the first application/client loads it then for every subsequent application it shares the memory pages of the DLL with their process address space. Thus, there is no unnecessary memory overload in the system.

□ 1.5x2 Marks

- (b) List **four (04)** differences between **RMI** and **CORBA**.

(6 marks)

ANSWER IN THIS BOX

- RMI is a Java-specific technology. CORBA has implementation for many languages.
- RMI uses Java interface for implementation. CORBA uses Interface Definition Language (IDL) to separate interface from implementation.
- RMI objects are garbage collected automatically. CORBA objects are not garbage collected because it is language independent and some languages like C++ does not support garbage collection.

- RMI programs can download new classes from remote JVM's. CORBA does not this code sharing mechanism.
- RMI passes objects by remote reference or by value. CORBA passes objects by reference.
- Java RMI is a server-centric model. CORBA is a peer-to-peer system.
- RMI uses the Java Remote Method Protocol as its underlying remoting protocol. CORBA use Internet Inter- ORB Protocol as its underlying remoting protocol.
- In RMI, the responsibility of locating an object implementation falls on JVM. The responsibility of locating an object implementation falls on Object Adapter either Basic Object Adapter or Portable Object Adapter.

→ 1.5x4 marks

- (c) Suppose you are developing a computerized train ticket issuing system for Sri Lanka Railway Department. Each ticket issuing counter of every train station will have one computer which runs a train ticket issuing client. Based on the transactions taking place at each client, the centralized database of this system will be updated in real time. The centralized database runs on a server which is located at the Sri Lanka Railway Head Quarters.

The system is supposed to have following features:

- Keeping records of the train schedules.
- Issuing, modifying and cancelling tickets.
- In case of train cancellation, re-allocating issued tickets to the next available train.
- Generating reports.
- User Management with a few different user levels.
- Automatically backing up the centralized database into a cloud storage at 11.30 PM every day.

- i. Draw a **high-level software architecture diagram** for the proposed system. Identify and label key hardware and software components.

(8 marks)

ANSWER IN THIS BOX

Students should identify the key components of the system such as ticket issuing clients, server(s), centralized database, and cloud.

They should also illustrate how communication should take place between these elements.

These components and communication should be modelled using a suitable middleware architecture they have learned.

Based on the correctness and completeness → 8 marks

- ii. Identify **suitable technologies** that could be used to develop each software component of this system. Briefly explain your **reasons** for the technologies selected.

(5 marks)

ANSWER IN THIS BOX

Students should identify suitable technologies for critical components such as databases, ticket issuing system, DBMS, cloud, etc.

Based on the correctness and completeness → 5 marks

- 4) The following piece of code was extracted from a Model class in a Restful backend application.

- (a) Explain the **functionality of the code** given below, while describing the **keywords in bold**.

(5 marks)

```
@Entity
@Table(name = "tbl_user")

public class Comment implements Serializable{
    @Column(name="first_name",length = 64)
    private String firstName;
    ...
    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String comment) {
        this.firstName = comment;
    }
    ...

    @Override
    public String toString() {
        JSONObject jsonObject = new JSONObject(this);
        return jsonObject.toString();
    }
    ...
}
```

ANSWER IN THIS BOX

The above code can be seen as a Model class written for a Java middleware application. → 1 mark

Purpose of this class is to bind table columns of tbl_user with instance variables **of the class** (e.g. instance variable '**private String firstName**' is mapped to the column '**first_name**'). → 2 marks

Getter and Setter methods are used to read and modify state of the object. → 1 mark

'toString()' method handles the serialization process (which returns a JSON object). This will enable the persisting and communicating capabilities for Comment objects. → 1 mark

- (b) State the functionality of following **HTTP Verbs**.

(4 marks)

ANSWER IN THIS BOX

POST:

Use POST to create new resources / data in the server. E.g. inserting a new record to a table → **1 marks**

PATCH:

PATCH is used to update a certain field of a record (data) in the server side. → **1 mark**

HEAD:

HEAD is used to obtain the header information of a particular web resource. → **1 mark**

GET:

Use GET requests to retrieve resource representation/information. → **1 mark**

- (c) The following piece of code was extracted from a Controller class in a Restful backend application.

```
@RequestMapping(value="user/updateUserId/{token}/{newEmailAddress}",
method=RequestMethod.PUT)
public void updateUserId(@PathVariable String token,
                        @PathVariable String newEmailAddress,
                        HttpServletResponse response) {
    .....
    .....
}
```

- i. Explain the meaning or functionality of **the words in bold**.

(8 marks)

@RequestMapping: This annotation will map a particular Java method to particular HTTP request URL. Once the URL is called, the method will be triggered automatically.

→ **2 marks**

method=RequestMethod.PUT: This defines the Type of the HTTP method which will be used for the communication between client and the server. In Restful applications PUT methods are used as update operations.

→ **2 marks**

@PathVariable: These are URL request variables that deliver the required information for the server method from the client.

→ **2 marks**

HttpServletResponse response: This is the initialization of the response message which will be send to the client from the server.

→ 2 marks

ii. What could be the overall functionality expected of the above method?

(2 marks)

[ANSWER IN THIS BOX]

Update the current email address with the new email address which is being currently used as the user id.

→ 2 marks

(d) The following piece of code was extracted from a Controller class of a Restful backend application.

```
Image image = imageService.findImageById(imageId);
```

i. In the context of a MVC application, what is the general functionality of a **Service** class?

(3 marks)

[ANSWER IN THIS BOX]

Service classes implements the business logic (methods to create, retrieve, and manipulate data) for a particular entity e.g., User Service will provide the creating, updating, deleting, and retrieving methods for users.

→ 3 marks

ii. In the context of a MVC application, what is the general functionality of a **Model** class?

(3 marks)

[ANSWER IN THIS BOX]

A Modal class provides the database operations required for the Service classes.

→ 3 marks
