

**UCSC****University of Colombo, Sri Lanka***University of Colombo School of Computing***DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY
(EXTERNAL)**Academic Year 2025 — 3rd Year Examination — Semester 6**IT6505 (R) — Middleware Architecture
(Repeat Paper)***Structured Question Paper*
(2 Hours)**To be completed by the candidate****Index Number**

--	--	--	--	--	--	--

Important Instructions

- The duration of the paper is **2 hours**.
- The medium of instructions and questions is English. Students should answer in the medium of English language only.
- This paper has **4 questions** on **12 pages**. Answer **all** questions.
- All questions carry **equal** marks.
- Write your answers **only on the space provided** on this question paper.
- Do not tear off any part of this question paper. Under no circumstances may this paper (or any part of this paper), 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 or part of a page is not printed, please inform the supervisor/invigilator immediately.
- Any electronic device capable of storing and retrieving text, including electronic dictionaries, smartwatches, and mobile phones, is not allowed.
- Calculators are **not allowed**.
- *All Rights Reserved*. This question paper can NOT be used without proper permission from the University of Colombo School of Computing.

**To be completed by
the examiners**

1	
2	
3	
4	
Total	

1)

- (a) Differentiate between **syntactic transparency** and **semantic transparency**.

(6 marks)

ANSWER IN THIS BOX

- (b) List and briefly explain the following *four (04)* **non-functional requirements** of distributed systems.

(8 marks)

ANSWER IN THIS BOX

1. **Scalability** –

2. **Openness** –

3. **Heterogeneity –**

4. **Fault tolerance –**

(c) List and explain *three (03)* types of **failure models**?

(6 marks)

ANSWER IN THIS BOX

1. **Halting failure –**

2. **Send-omission failure –**

3. **Byzantine failure –**

- (d) Why are **distributed systems** considered less reliable than **centralized systems**? Discuss at least four (04) points.

(5 marks)

ANSWER IN THIS BOX

- 2)
(a) Explain the concept of **integration infrastructure**.

(5 marks)

ANSWER IN THIS BOX

- (b) Explain the use of **protocols** in middleware.

(5 marks)

ANSWER IN THIS BOX

- (c) Explain the meaning of marshalling. Describe the importance of this operation.

(6 marks)

ANSWER IN THIS BOX

- (d) List *four (04)* actions performed by a **message broker**.

(4 marks)

ANSWER IN THIS BOX

- (e) Explain the **publish - subscribe** model.

(5 marks)

ANSWER IN THIS BOX

- (c) Discuss *two (02)* reasons why **CORBA** did not gain widespread adoption?

(4 marks)

ANSWER IN THIS BOX

- (d) What is meant by a **container** in transactional middleware.

(4 marks)

ANSWER IN THIS BOX

- (e) Describe how RMI makes **remote calls**.

(5 marks)

ANSWER IN THIS BOX

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

4.

(a) Explain the steps involved in creating an **RMI** application?

(5 marks)

ANSWER IN THIS BOX[illegible]

(b) Identify the most suitable **HTTP verb** to be used in each of the following RESTful URIs.

(5 marks)

ANSWER IN THIS BOX

URI	HTTP Method
/teaOrder/{id}	
/teaOrder/add	

/teaOrder/delete/{id}	
/teaOrder/getAll	
/teaOrder/update/{id}	

- (c) The following piece of code was taken from a *Controller class* in a RESTful backend application designed for a Courier Management System.

Explain The *functionality* of the code given below.

(10 marks)

```
package com.example.courierapp.controller;

import com.example.courierapp.model.Parcel;
import com.example.courierapp.service.ParcelService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/parcels")
public class ParcelController {

    @Autowired
    private ParcelService parcelService;

    @PostMapping
    public ResponseEntity<Parcel> createParcel(@RequestBody Parcel parcel) {
        Parcel createdParcel = parcelService.createParcel(parcel);
        return ResponseEntity.ok(createdParcel);
    }

    @GetMapping
    public ResponseEntity<List<Parcel>> getAllParcels() {
        return ResponseEntity.ok(parcelService.getAllParcels());
    }

    @GetMapping("/{id}")
    public ResponseEntity<Parcel> getParcelById(@PathVariable Long id) {
        Parcel parcel = parcelService.getParcelById(id);
        return parcel != null ? ResponseEntity.ok(parcel) :
        ResponseEntity.notFound().build();
    }

    @PutMapping("/{id}")
    public ResponseEntity<Parcel> updateParcel(@PathVariable Long id,
        @RequestBody Parcel
        updatedParcel) {
        Parcel parcel = parcelService.updateParcel(id, updatedParcel);
```

```
        return parcel != null ? ResponseEntity.ok(parcel) :
        ResponseEntity.notFound().build();
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteParcel(@PathVariable Long id) {
        boolean deleted = parcelService.deleteParcel(id);
        return deleted ? ResponseEntity.noContent().build()
            : ResponseEntity.notFound().build();
    }
}
```

ANSWER IN THIS BOX

- (d) The following service call was found in a piece of code in a **Controller class** of a Restful backend application.

```
Order order = orderService.findOrderById(orderId);
```

Briefly explain the expected functionality of the above *service call*.

(5 marks)

ANSWER IN THIS BOX
