Name: Class:

JURONG PIONEER JUNIOR COLLEGE

JC2 Year – End Examination 2025

COMPUTING **9569/01**

Higher 2 **17 September 2025**

Paper 1 (Written) **3 hours**

Additional materials: 12-page Answer Booklet

Answer **all** the questions.

Approved calculators are allowed.
You are reminded of the need for clear presentation in your answers.

Answer papers will be provided with the question paper.
Write your name and civics class on all the work that you hand in.
Write in **dark blue** or **black pen** on both sides of the paper.
You may use an HB pencil for any diagrams or graphs.
Do not use staples, paper clips, glue or correction fluid.

Write your answers to every question **on a fresh page of paper**.
Approved calculators are allowed.

The number of marks is given in brackets **[ ]** at the end of each question or part question.
The total number of marks for this paper is **100**.

This document consists of **9** printed pages and **3** blank pages.

[Turn over

BLANK PAGE

1 JPCAR operates a car-sharing network across Singapore. Users can rent cars from docking stations using a mobile app. The company maintains two types of data storage:

Relational Database (RDBMS): Stores structured data like user accounts, cars, stations, and rentals.

NoSQL Database (Document Store): Stores unstructured data like real-time Global Positioning System (GPS) coordinates of bicycles and usage logs from Internet of Things (IoT) devices.

Anyone who wishes to use the car-sharing network needs to sign up to be a user of the JPCAR. The company wants to store data in a relational database with the following requirements:

- Each user has a unique `userID` that corresponds to a `name`, `email`, and `phoneNo`.
- Each car has a `carID` that is linked to a `stationID` (where it is parked), and `status` (`Available` or `In Use`).
- Each station has a `stationID` that corresponds to a `stationName`, and `location`.

It is given that whenever a user rents an available car parked at the station, the car's status will be set to "`In Use`" immediately, and the rental date and rental start time will be recorded by the system. Upon returning, the returned car's status will be changed to "`Available`". Rental end time, total rental cost, and information on the station the car was returned to will also be determined and recorded.

(a) JPCAR wants to use a relational database to store and manage the data for the system.

(i) A database requires several tables to store the data.

Draw an entity-relationship (ER) diagram to show the tables in third normal form (3NF) and the relationship(s) between them. [2]

(ii) A table description can be expressed as:

```
TableName (Attribute1, Attribute2, Attribute3, …)
```

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using an asterisk (*).

Write table descriptions for the tables identified in part **(a)(i)**, using the information given. [4]

(b) Using tables created, write SQL statements for the following queries:

(i) display all the available cars parked at the location "`Desker Road`". [2]

(ii) display every `userID`, and the total number of car rentals the user has made. [2]

(iii) display the top **three** users who spent the highest total cost on car rentals, sorted from highest to lowest. [2]

(c) Explain how double-booking conflicts on a particular car between **two** or more users can be prevented. [2]

(d) Explain **two** key differences between a backup file and an archive file. [2]

(e) The real-time locations of cars are stored in a NoSQL document database (e.g., MongoDB) using the following document structure:

```
{
 "carID": "C1001",
  "location": { "lat": 1.283, "lon": 103.860 },
  "battery%": 85,
  "lastUpdate": "2025-07-15T13:45:00Z"
}
```

(i) Write a MongoDB query to find all bicycles with battery < 20% [2]

(ii) Explain **one** advantage of using NoSQL over SQL for storing GPS tracking data. [2]

(iii) Describe **one** scenario where the relational database design is still preferable for JPCAR. [2]

2 A start-up company is developing a ride-hailing application called JRide. The system will have different types of vehicles, such as car and motorbike, that share common attributes and behaviours but also have some unique features. The company wants to use Object-Oriented Programming (OOP) to model this situation.

The system needs to:

- Store details of the driver's name, vehicle registration number, and maximum passenger capacity.
- `Motorbike` can only carry one passenger while `Car` can carry up to seven passengers.
- The fare is calculated differently depending on the type of vehicle. For example, `Car` calculates fare based on distance and a base fee, while `Motorbike` calculates fare based only on distance with a lower rate.

(a) Draw a class diagram for the described situation, showing:

- any derived classes and inheritance from the base class
- the properties needed in the base and any derived classes
- suitable methods, in each class, to support the system. [6]

(b) The company has decided to introduce a `Driver` class to store details of the driver's name, license number, and years of experience. Describe how you would update the class diagram to include this new class. [3]

It is common for the properties of a class to be private.

(c) Explain what encapsulation means in OOP. [2]

(d) Describe how it could be applied to this situation. [2]

3 A college is launching an online e-Assessment portal. Teachers record the marks in spreadsheets and upload them to the portal. The spreadsheets must be verified for both authenticity and integrity. Additionally, the portal must be protected from the public Internet.

(a) Explain how a digital signature works in this situation. [5]

(b) State one key difference between encrypting a file and digitally signing a file. [1]

(c) Propose a firewall policy to protect the e-Assessment server that is reachable over HTTPS from the Internet but administered only from the IT office. [2]

4 A school maintains a Binary Search Tree (BST) of locker IDs to support quick allocation, deallocation and look-ups. Each node stores a unique integer key (the locker ID), and pointers left and right.

Unless stated otherwise, assume no duplicate keys and that the BST property holds: all keys in the left subtree < key, all keys in the right subtree > key.

(a) The following locker IDs are inserted into an initially empty BST in the following order: `42, 18, 60, 12, 35, 50, 70, 63, 72`

(i) Draw the resulting BST. [2]

(ii) Write the in-order traversal output of your tree as a comma-separated list. [1]

Assume each node has fields: `key`, `left`, `right`.

(b) Write a **recursive** pseudocode for `InOrder(root)` that displays the keys in ascending order. [4]

(c) Describe an **iterative** algorithm for `BST_Search(root, k)` that returns `TRUE` if found or `FALSE` if not found. [5]

(d) State and justify the worst-case time complexity of searching in a BST using an iterative algorithm. [2]

(e) Explain why you would store the ordered locker IDs in a BST rather than an ordered array. [2]

The school's allocation service buffers locker assignment requests in a fixed-capacity circular queue before processing them against the BST. The queue is implemented as an array that stores up to 50 integers.

(f) Write the pseudocode for `DEQUEUE()` which returns the integer stored at the front of queue or -1 if the queue is empty. [6]

(g) Explain one advantage of a circular queue over a linear queue in this situation. [2]

5 A smart thermostat stores the current temperature setpoint and shows a short history of recent changes on its display.

The following describes the implementation:

Variables

- `SETPOINT` an integer temperature
- `HISTORY` a list of change records kept in memory that expands as needed
- `DISPLAY_BUFFER` text that is used to draw the screen

On start-up

1. Set `SETPOINT` to 24
2. Create an empty `HISTORY` list

Main loop (every 200 milliseconds)

3. If the dial is turned, update `SETPOINT` to the dial value
4. Add a new change record (containing the current `SETPOINT` and time) to `HISTORY`.
5. Initialise a 32-character `DISPLAY_BUFFER`, format the current setpoint and a few historical items, display it on screen.
6. Repeat from step 3.

After several months in service, many units begin to reset and eventually refuse to start, showing “out of memory”.

(a) Explain how mistakes in the algorithm caused this recurring problem. [2]

(b) Explain how the algorithm should be changed to prevent the problem from recurring. [2]

(c) State one advantage and one disadvantage of using dynamic memory allocation (compared with static allocation) in embedded devices like this thermostat. [2]

6 A college hosts a web application on a web server at the domain `portal.jpjc.edu.sg`. Students access it from home and on campus.

(a) Explain how a client obtains the IP address for `portal.jpjc.edu.sg` using a Domain Name Server (DNS). [3]

(b) Data transmitted across the Internet is divided into sequentially numbered packets.

(i) State two reasons why data is split into packets for transmission across the Internet. [2]

(ii) State two reasons why the packets are numbered. [2]

(iii) State two items, other than the packet number, that are stored in the packet header. [2]

(c) State the key difference between a static IP address and a dynamic IP address. [1]

(d) For the 300 student laptops connecting to the campus Wi-Fi, recommend either static or dynamic addressing and justify your choice. [3]

7 A junior college is rolling out an online admissions portal that asks applicants for their full name, email, mobile, NRIC, CCA, medical condition (optional), emergency contact, bank account (for bursary), and uploads of a passport-style photo.

(a) Define the term “personal data” and give one example from this situation. [2]

(b) State a possible threat of losing the applicants’ data and suggest a method to prevent it. [2]

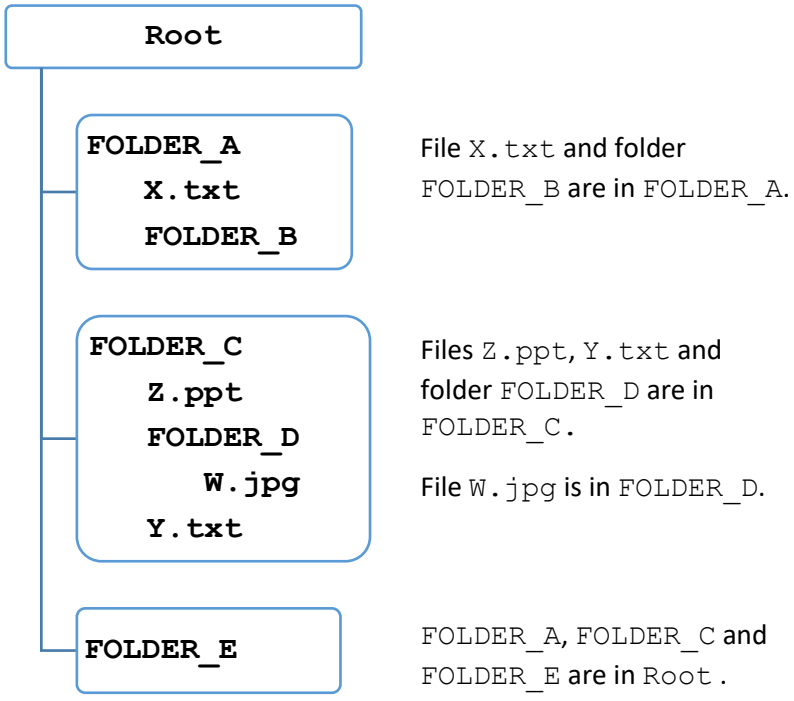
(c) Recommend two Personal Data Protection Act (PDPA) controls the college should implement for the portal. [2]

8 A media server scans a nested folder structure to list all files. Each folder may contain files and subfolders. The following is a recursive procedure:

```
PROCEDURE ListAll(f: FOLDER)
    IF f <> NULL AND f.isEmpty() <> TRUE THEN
        FOR EACH item IN f.items DO
            IF item.type = "FILE" THEN
                OUTPUT item.name
            ELSE // item is a FOLDER
                ListAll(item) // recursive call
            ENDIF
        ENDFOR
    ENDIF
ENDPROCEDURE
```

(a) Describe the base case. [2]

(b) The call `ListAll(Root)` is performed on the following structure:



(i) List the recursive calls made for the call `ListAll(Root)` in the exact order they occur. [2]

(ii) Write the exact sequence of file names displayed by the procedure. [1]

(iii) State the number of times `ListAll` called itself. [1]

(c) Explain when and why a memory stack overflow could occur with `ListAll`. [2]

END OF PAPER

BLANK PAGE

BLANK PAGE