

NANYANG JUNIOR COLLEGE  
JC2 PRELIMINARY EXAMINATIONS  
Higher 2

---

# COMPUTING

**9569/01**

Paper 1 Written

**15 September 2026**

**3 hours**

---

## READ THESE INSTRUCTIONS FIRST

An answer booklet will be provided with this question paper. You should follow the instructions on the front cover of the answer booklet. If you need additional answer paper ask the invigilator for a continuation booklet.

Answer **all** questions.

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.

- 1 A wildlife rescue centre is developing a program to manage the animals under its care. Birds, mammals, and reptiles are cared for at the centre.

For all animals, the data that will be stored include:

- Animal ID
- Name of the animal
- Age of the animal
- Health status (Healthy/Sick)
- Weight of the animal (in kg).

For birds, the additional data stored include:

- Wingspan
- Flighted (whether the bird is capable of flight - True/False)
- Seed proportion
- Insect proportion.

For mammals, the additional data stored include:

- Body temperature
- Nocturnal (True/False)
- Number of daily feeding sessions.

For reptiles, the additional data stored include:

- Preferred temperature
- Venomous (True/False)
- Food serving temperature.

When managing an animal:

- health status can be retrieved or updated
- daily food requirement is calculated based on the animal's weight and a given body condition score (measured on a scale of 1 to 9).

When food requirements are calculated:

- for a bird, it takes into consideration the seed and insect proportions
- for a mammal, it takes into consideration the number of daily feeding sessions
- for a reptile, it takes into consideration the food serving temperature.

Object-oriented programming will be used to model the animals.

- (a) Draw a class diagram that shows the following for the system described above.
- the superclass
  - any subclasses
  - inheritance
  - properties
  - appropriate methods. [8]
- (b) Explain how encapsulation prevents an animal's health status or weight from being inadvertently modified. [2]
- (c) Explain how polymorphism allows more animal types to be added to the program without changing the implementation of animal feeding. [2]
- A developer suggests moving the `Wingspan` data into the base `Animal` class so it can be accessed by all animal subclasses.
- (d) Suggest why this is a poor application of the inheritance principle. [2]

- 2** A satellite sends a calibration signal that lasts for a specified number of milliseconds. The signal is constructed using:

- a short pulse lasting 2 milliseconds
- a long pulse lasting 5 milliseconds.

The order of the pulses matter.

The recursive function `CountSignals()` returns the number of different signals that can be constructed for a given duration (in milliseconds).

```

01 FUNCTION CountSignals(Duration: INTEGER) RETURNS INTEGER
02
03     IF Duration = 0 THEN
04         RETURN 1
05     ELSE IF Duration < 2 THEN
06         RETURN 0
07     ENDIF
08
09     RETURN CountSignals(Duration - 2) + CountSignals(Duration - 5)
10
11 ENDFUNCTION

```

- (a)** Draw a complete trace tree for the function call `CountSignals(7)`.

Your trace tree should show:

- the parameter value for every function call
- the value returned by every function call. [3]

- (b)** Explain what might happen if the required signal duration becomes exceedingly large, during the execution of the recursive algorithm. [2]

- (c)** Write an iterative function, `CountSignalsIterative()`, that returns the same result as `CountSignals()`.

You may use a one-dimensional array in your implementation. [4]

- (d)** Compare the performance of the recursive and iterative algorithms as the required signal duration increases. [2]

- 3 A nationwide online retailer, Delivery Zone, processes customer orders every day.

The following array of values stored in an array are to be sorted into ascending order.

42 17 68 25 10 55 31 49

For this question, you may assume that the quicksort always selects the first element (left end) of each sub-array as the pivot.

- (a) Using annotated diagrams, show how quicksort algorithm sorts the values in the array into ascending order. [3]
- (b) Using annotated diagrams, show how merge sort algorithm sorts the values in the same array into ascending order. [3]

The retailer handles two different processes, with specific requirements:

- **Process A**

The retailer makes a daily archival of 12 million orders, to be sorted within 10 minutes.

- **Process B**

Once a day, the list of orders (already sorted by order number) are handed over to the warehouse for processing. At the warehouse, the orders need to be re-sorted according to order value.

A programmer is deciding whether to use quicksort or merge sort for each process. The programmer claims that "*Quicksort is always more efficient than Merge Sort!*"

- (c) Discuss whether the programmer's statement is valid. [3]
- (d) Explain which of the two sorting algorithms is more suitable for each process. You are not allowed to suggest the same algorithm for both processes.

Justify your answer by referring to the characteristics of the sorting algorithm and the requirements of the respective processes. [4]

- (e) Apart from time complexity, state **one characteristic** of merge sort that may make it more suitable than quicksort in certain applications. [1]

- 4 A Pokemon Trainer is developing an application to store the names of Pokemon that have been caught.

The names of Pokemon that have been caught are stored in a binary search tree implemented using an array, `pokemon_bst`. Each element of the array contains three values. `pokemon_name` is a string, and `left_pointer` and `right_pointer` are integers. A pointer value of -1 indicates the null pointer.

The variable `root` stores the array index of the root of the binary search tree.

The contents of the array `pokemon_bst` is shown below.

|                                                          |      |          |            |              |              |               |
|----------------------------------------------------------|------|----------|------------|--------------|--------------|---------------|
| <table><tr><td>root</td></tr><tr><td>2</td></tr></table> | root | 2        | Index      | left_pointer | pokemon_name | right_pointer |
|                                                          | root |          |            |              |              |               |
|                                                          | 2    |          |            |              |              |               |
|                                                          | 0    | -1       | Bulbasaur  | -1           |              |               |
|                                                          | 1    | -1       | Eevee      | -1           |              |               |
|                                                          | 2    | 0        | Charmander | 3            |              |               |
|                                                          | 3    | 1        | Pikachu    | 4            |              |               |
| 4                                                        | -1   | Snorlax  | 5          |              |              |               |
| 5                                                        | -1   | Squirtle | -1         |              |              |               |

- (a) Draw the binary search tree represented by the data in the array `pokemon_bst` and the value in `root`. [2]
- (b) A new Pokemon, `Dragonite`, is caught and inserted into the `pokemon_bst`. Copy and complete the table above to show any changes made to the array after this insertion. [2]
- (c) The application stores the same Pokemon names in both a binary search tree and a hash table. Explain the benefits of storing the Pokemon names in two different data structures instead of only one. In your answer, include one example of an operation that is better suited to each data structure. [3]

- (d) The same Pokemon names are stored in an initially empty hash table with 10 locations, indexed from 0 to 9.

The procedure `insert_pokemon` below is used to insert a Pokemon into the hash table.

```

01  PROCEDURE insert_pokemon(hash_table, Pokemon)
02
03      // ASCII(char) returns the ASCII value of char as integer
04      hash_value ← ASCII(Pokemon[0]) MOD 10
05      index ← hash_value
06      count ← 0
07
08      WHILE hash_table[index] <> "" AND count < 10
09          index ← (index + 1) MOD 10
10          count ← count + 1
11      ENDWHILE
12
13      IF count < 10 THEN
14          hash_table[index] ← Pokemon
15      ENDIF
16
17  ENDPROCEDURE

```

- (i) Describe the collision resolution method used by the procedure. [2]
- (ii) The Pokemon are inserted into the hash table in the following order:

| Pokemon    | First letter | ASCII |
|------------|--------------|-------|
| Charmander | C            | 67    |
| Pikachu    | P            | 80    |
| Bulbasaur  | B            | 66    |
| Snorlax    | S            | 83    |
| Eevee      | E            | 69    |
| Squirtle   | S            | 83    |
| Dragonite  | D            | 68    |

Following the procedure shown above, copy and complete the hash table after all seven Pokemon have been inserted. [2]

| index | pokemon_name |
|-------|--------------|
| 0     |              |

- (iii) A new Pokemon, `Blastoise`, is caught. Copy and complete the trace table below to show the program state after each iteration:

`insert_pokemon(hash_table, "Blastoise")`

| hash_value | index | count | hash_table[index] | WHILE condition (True/False) |
|------------|-------|-------|-------------------|------------------------------|
|            |       |       |                   |                              |

[4]

- (iv) Explain why the variable `count` is required for the procedure to operate correctly. [2]

- 5 An enterprise software company, DevLogix, tracks system bugs and feature requests submitted by corporate clients.

The platform uses a database to store clients' support tickets for the specific software modules it maintains. It also stores details of the internal development teams assigned to fix them. Internal teams are specialised; no two development teams maintain the same software module.

Each client has:

- a client ID
- a company name
- a corporate headquarters address
- a technical contact email address.

Each development team has:

- a team code
- a team name
- a lead engineer's email address
- an internal phone extension number.

Each software module has:

- a module identifier
- a description
- a core programming language
- a total active bug count
- a critical threshold bug limit.

Each support ticket has:

- a ticket number
- an open date
- a set of ticket items, where each ticket item specifies the priority and estimated fixing hours for a specific software module associated with the client's ticket.

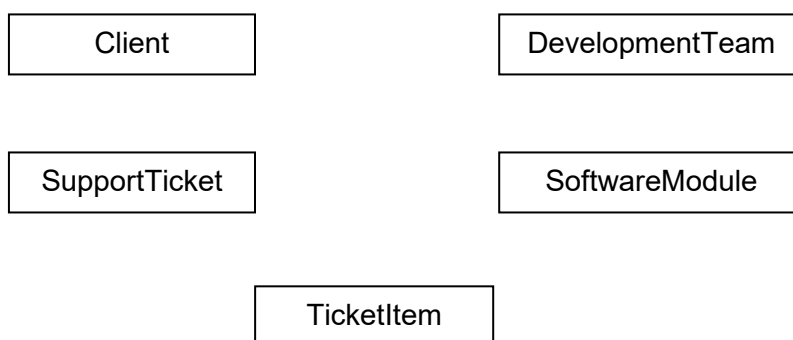
The engineering department currently logs new activity into a centralised, flat tracking table as shown:

**TicketLog:**

| TicketNo | OpenDate   | ClientID | ClientName | ModuleID | Priority | EstHours | TeamName      |
|----------|------------|----------|------------|----------|----------|----------|---------------|
| T1001    | 2026-11-01 | C801     | Nexus Corp | MOD-AUTH | High     | 8        | Security Team |
| T1001    | 2026-11-01 | C801     | Nexus Corp | MOD-BILL | Medium   | 4        | Finance Devs  |
| T1002    | 2026-11-02 | C402     | Apex Logic | MOD-DASH | Low      | 3        | UI-UX Squad   |
| T1003    | 2026-11-03 | C801     | Nexus Corp | MOD-AUTH | High     | 12       | Security Team |

- (a) State the primary key used to uniquely identify each row in the **TicketLog** table. [1]
- (b) Normalisation is a process used when designing database tables.
- (i) Explain why the table **TicketLog** is not in second normal form (2NF). [2]
- (ii) State **one** drawback of the table **TicketLog** not being in third normal form (3NF). [1]
- (c) The normalised database will have **five** tables.

Copy and complete this entity-relationship (ER) diagram by drawing **four** many-to-one relationship lines connecting the entity blocks. When copying the diagram, do **not** rearrange the layout of the five given entities. If necessary, your connecting lines may cross over one another.



[4]

- (d) A table description can be expressed as:

TableName (Attribute 1, Attribute 2, Attribute 3, ...)

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline.

Assume that the system has been properly normalised to 3NF. Write the complete table descriptions **only** for the **Client**, **SupportTicket** and **TicketItem** tables. Do not use any compound primary keys. [5]

- (e) Write an SQL query to retrieve the ticket number, open date, company name, and the estimated fixing hours for all items associated with the module 'MOD-AUTH' opened after November 1, 2026 (2026-11-01). The output must list the records in descending order of the estimated fixing hours. [5]
- (f) State **two** challenges the company would face if they attempted to migrate from the current relational database system to a NoSQL database management system. [2]

- 6** A telemedicine company allows patients to consult doctors via video call, receive electronic prescriptions, and access their medical records through a mobile application. The company also deploys its application in multiple countries. All data is transmitted over the internet between patients' devices, the telemedicine platform servers, and partner pharmacies.

- (a)** The platform uses Unicode encoding to transmit patient data. A patient's unique identifier code begins with the character **N** whose hexadecimal value is **4E**.
- (i)** Convert the hexadecimal value **4E** to its denary equivalent. [1]
  - (ii)** Explain why the platform uses Unicode encoding and not ASCII. [2]
- (b)** The data is transmitted across the internet using a packet-switching network.
- (i)** Explain what is meant by a packet-switching network. [2]
  - (ii)** Explain why packets are numbered sequentially when a prescription message is transmitted from the telemedicine server to a pharmacy. [2]
  - (iii)** State and explain one disadvantage of using a packet-switching network in this scenario. [2]

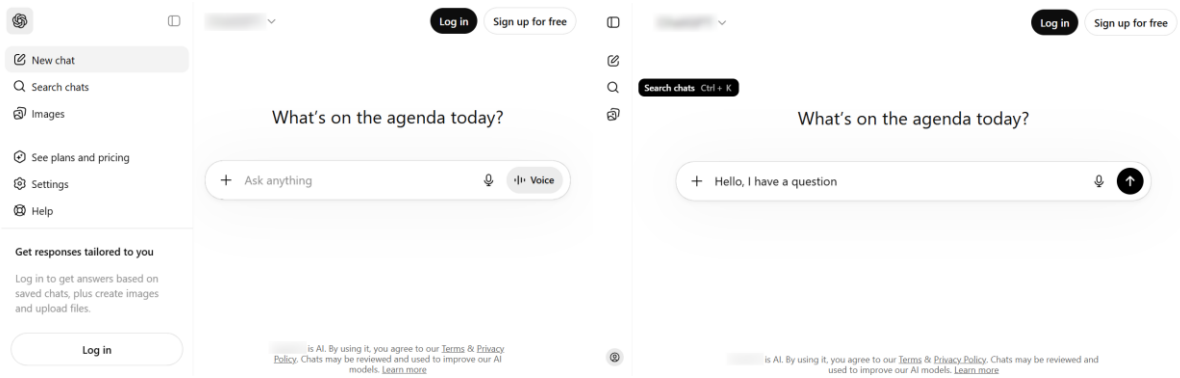
The platform's IT team is investigating a reported incident in which an attacker intercepted a prescription message in transit and altered the dosage instructions before it reached the pharmacy.

- (c)** Explain how a digital signature and a digital certificate could have been used to detect this attack. [4]

Recently, the company's servers experienced a sudden surge of connection requests from thousands of different IP addresses, causing it to become unresponsive to legitimate patients' requests.

- (d)**
- (i)** Identify the type of attack described above. [1]
  - (ii)** An IT intern suggested deploying a firewall in defending against this type of attack. Discuss the effectiveness of the intern's suggestion. [2]

7 Some screenshots from the web interface of a leading AI chat platform are shown.



- (a) With reference to the screenshots, describe two ways in which usability principles were applied in the design of this interface. [2]
- (b) State two obligations that the AI chat platform must comply with under the terms of the Personal Data Protection Act (2012) regarding the use and storage of personal data. [2]
- (c) Explain two ethical considerations surrounding the use of AI chat platforms. [4]

-- END OF PAPER --