

APOLLO

Computer Science HL Workbook

First assessment 2027

Whole questions arranged by syllabus topic.

Original question and scheme crops preserve source diagrams.

APOLLO

A1 Computer fundamentals

1 questions

APOLLO

Shared case study

Read this source alongside the questions that refer to it.

Computer science

Case study: Preventing network security attacks

Specimen paper

Instructions for candidates

- Case study booklet required for higher level and standard level paper 1 computer science examinations.

Introduction

The integration of technology in healthcare has transformed how medical services are delivered. However, with this increased use of technology, healthcare institutions like private hospitals have become attractive targets for cybercriminals and therefore face significant security challenges that could have devastating impacts. This case study highlights the interventions used to prevent security incidents by one private hospital, TaylorMade Healthcare, and explores some network security practices.

Background

TaylorMade Healthcare uses high-speed internet to operate its network infrastructure, which includes computers, servers, switches, routers, wireless access points, and *firewalls*. These devices and systems are connected internally and to the internet, enabling staff to access medical records, communicate with other users and patients, exchange information, and transfer data. The devices and systems work cohesively and are vital for the hospital's smooth operation, including patient care, data management, and resource handling.

TaylorMade Healthcare stores vast amounts of sensitive data, such as patient health records, financial information, and confidential documents, which it must safeguard to maintain trust and confidence among its patients, staff, and stakeholders.

The hospital's director acknowledges the importance of securing the network and data against risks like *malware* infections, *phishing* attacks, and unauthorized access. The director also knows that it is critical to implement effective security measures to protect the hospital's network infrastructure and data.

Bring-your-own-device (BYOD) policy

TaylorMade Healthcare has adopted a bring-your-own-device (BYOD) policy that permits staff to use personal electronic devices for work tasks. Although a BYOD policy can provide many benefits, such as increased flexibility and convenience, it also has several weaknesses.

Ransomware and rootkit attacks

TaylorMade Healthcare recently suffered a ransomware attack that targeted its network infrastructure. The attack was launched through a phishing email that appeared to be from a trusted source. When an unsuspecting staff member clicked on the link in the email, the ransomware gained access to the hospital's network and quickly spread, encrypting sensitive data and rendering it inaccessible.

The consequences of the attack were severe, leading to significant data loss and system downtime. The ransomware attack encrypted patients' data and prevented staff from accessing patient records for several days. This led to substantial disruptions.

TaylorMade Healthcare's reputation has suffered due to the attack, with patients expressing concern about the security of their data. There was also a significant financial impact, with TaylorMade Healthcare paying the attackers a ransom to regain access to its encrypted data.

Another notable concern is a rootkit attack, which uses malicious software to hide itself within an operating system to gain administrative-level access to the network. Rootkits are notoriously difficult to remove.

Interventions

45 “Dealing with the ransomware attack was a wake-up call for our entire team,” said Jane Wang, Head of Cybersecurity. “We realized the importance of being proactive and creating a culture of security awareness,” she added, “and we also realized that a comprehensive security strategy is essential for protecting sensitive data and networks.”

Since the attack, TaylorMade Healthcare has identified several measures to prevent future attacks:

- Data backup and *disaster recovery plans*
- *Email filtering*
- *Failover systems*
- 50 • Firewall implementation
- Network segmentation and *access control*
- Regular software and security updates
- *Third-party services*
- *User awareness training*

55 Implementing network segmentation and access control reduced the risk of cyber-attacks and improved the security of TaylorMade Healthcare’s network. These and the other interventions have contributed to reducing the effects of security breaches, ensuring the confidentiality and integrity of sensitive information. These safeguards have also minimized system downtime and strengthened data protection, which is vital for the hospital’s uninterrupted functioning.

60 Challenges faced

TaylorMade Healthcare’s goal is to build resilience to address the following challenges.

Standard level and higher level:

- Develop strategies to create an awareness of cybersecurity, especially ransomware and rootkit attacks.
- 65 • Consider the benefits of a bring-your-own-device (BYOD) policy while preparing for its security weaknesses.

Higher level only:

- Explore the viability of network segmentation and access control.
- Develop procedures to restore operations in case of a breach.

Additional terminology

Access control
Bring your own device (BYOD)
Cybercriminal
Disaster recovery plan
Email filtering
Failover system
Firewall
Malware
Phishing
Ransomware
Rootkit attack
Third-party services
User awareness training

Some companies, products, or individuals named in this case study are fictitious and any similarities with actual entities are purely coincidental.

Section A

Answer **all** questions from this section. Answers must be written within the answer boxes provided.

1. A team of programmers use platform as a service (PaaS) via their laptops to develop a software application.

(a) Outline the function of an accumulator. [2]

.....

.....

.....

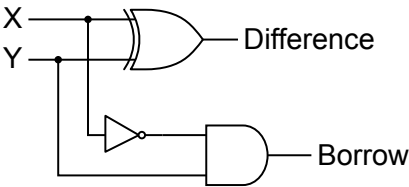
.....

(b) State the binary equivalent of the hexadecimal number AC. [1]

.....

.....

(c) Construct a truth table for the following circuit: [2]



X	Y	Difference	Borrow

(This question continues on the following page)





(Question 1 continued)

(d) Outline the role of accounting in an operating system. [2]

.....

.....

.....

.....

(e) Outline why platform as a service (PaaS) is a suitable environment in which to build applications. [2]

.....

.....

.....

.....

(f) Explain why the execution speed of compiled code is faster than that of interpreted code. [4]

.....

.....

.....

.....

.....

.....

.....

.....



Turn over

APOLLO

A2 Networks

2 questions

Section A

Answer **all** questions from this section. Answers must be written within the answer boxes provided.

2. A hospital has a local area network (LAN) connected to the internet to provide access to cloud services.

(a) State **one** network device used to connect the local area network (LAN) to the internet. [1]

.....

.....

The TCP/IP model was developed to facilitate reliable and robust communication over a network.

(b) Describe the function of the network interface layer of the TCP/IP model. [2]

.....

.....

.....

.....

A router uses a fiber-optic cable to connect directly to an internet service provider’s equipment.

(c) List **three** advantages of using a fiber-optic cable instead of a twisted-pair cable. [3]

.....

.....

.....

.....

.....

.....

(This question continues on the following page)





(Question 2 continued)

(d) Explain how a switch routes data packets across a network. [4]

.....

.....

.....

.....

.....

.....

.....

.....

The hospital deploys a hardware firewall incorporating network address translation (NAT).

(e) Explain **one** function of network address translation (NAT) in a firewall. [4]

.....

.....

.....

.....

.....

.....

.....

.....



Turn over

Section B

Answer the following question, which refers to the case study. Answers must be written within the answer boxes provided.

5. Ransomware and rootkit are types of malware.

(a) (i) Identify **two other** types of malware. [2]

.....

.....

.....

.....

A phishing attack caused ransomware to be installed on TaylorMade Healthcare’s network, resulting in financial loss.

(ii) Outline **one other** potential consequence of a phishing attack on a hospital. [2]

.....

.....

.....

.....

(This question continues on the following page)



(Question 5 continued)

A bring your own device (BYOD) policy allows employees to use their own devices to access data on TaylorMade Healthcare’s local area network (LAN) via the internet.

- (b) Outline **two** strategies that TaylorMade Healthcare could use to maintain security when employees use their own devices. [4]

.....

.....

.....

.....

.....

.....

.....

.....

- (c) Outline **two** advantages of a bring your own device (BYOD) approach compared with the hospital providing devices to employees. [4]

.....

.....

.....

.....

.....

.....

.....

.....

(This question continues on the following page)



Turn over

TaylorMade Healthcare identified several challenges to build cyber resilience.

- [12]



APOLLO

A3 Databases

1 questions

Section A

Answer **all** questions from this section. Answers must be written within the answer boxes provided.

3. A hospital stores patients’ data in a relational database.

The database view, VISITS, is an extract showing the dates patients visited the hospital for an appointment.

The VisitDate field represents the date of each patient’s appointment.

PatID	PatientName	PatientDOB	DocID	Doctor	VisitDate	Treatment
1	Samuel Moody	2007/06/11	1	Jill Smith	2025/01/11	Bone scan
1	Samuel Moody	2007/06/11	2	Norayr Lomidze	2025/01/12	Physical therapy
2	Lin Kobayashi	1984/02/24	1	Jill Smith	2025/01/12	Physical therapy
3	Yejoo Kim	1999/08/18	3	Phoebe Drakos	2025/01/13	Radiation therapy
3	Yejoo Kim	1999/08/18	3	Phoebe Drakos	2025/01/14	Therapy part 1

(a) Construct an SQL SELECT query to list the names of patients whose treatment includes the word “therapy”.

A patient’s name should only appear once. [3]

(This question continues on the following page)



— , —

(Question 3 continued)

The database view, VISITS, can be represented by the following notation:

```
VISITS (PatID, PatientName, PatientDOB, DocID, Doctor,
VisitDate, Treatment)
```

- (b) Construct the database in third normal form (3NF) for the three entities in the VISITS view extract. [5]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

The hospital has transitioned its database system to a distributed model. Atomicity can be applied across multiple nodes when transactions are executed.

- (c) Describe how atomicity manages transactions across multiple nodes. [3]

.....

.....

.....

.....

.....

.....

(This question continues on page 9)



Turn over



Please **do not** write on this page.

Answers written on this page
will not be marked.





(Question 3 continued)

The hospital’s research laboratory tracks virus outbreaks. A spatial database stores the addresses of patients who have contracted a virus.

- (d) Outline why a spatial database is used to store the addresses of patients who have contracted a virus.

[3]

.....

.....

.....

.....

.....

.....



Turn over

APOLLO

A4 Machine learning

1 questions

Section A

Answer **all** questions from this section. Answers must be written within the answer boxes provided.

4. A hospital uses a deep learning convolutional neural network (CNN) to analyse chest X-rays and predict the likelihood of a patient developing pneumonia.

(a) State **one** of the layers in a convolutional neural network (CNN). [1]

.....

.....

(b) Identify **one** reason data cleaning is carried out before training a convolutional neural network (CNN). [1]

.....

.....

(c) Outline **one** reason a graphics processing unit (GPU) is recommended when training a deep learning convolutional neural network (CNN) model using a large dataset. [2]

.....

.....

.....

.....

(d) Describe how the deep learning convolutional neural network (CNN) model could be evaluated using an F1 score. [3]

.....

.....

.....

.....

.....

.....

(This question continues on the following page)





(Question 4 continued)

The hospital purchased a self-learning robotic cleaner that uses reinforcement learning (RL).

- (e) Justify why a robotic cleaner would use reinforcement learning (RL) instead of supervised learning (SL). [3]

.....

.....

.....

.....

.....

.....

A deep learning algorithm was implemented to automatically monitor the hospital’s social media sites. The model uses natural language processing to classify comments into categories like “compliments” and “complaints”.

Members of the public are asked to remove any comments that are not supported by evidence. However, the hospital’s CEO is concerned about the ethical issues of doing this, as it may be considered online harassment.

- (f) Discuss whether the hospital should ask members of the public to retract their comments. [5]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Turn over

APOLLO

B1 Computational thinking

4 questions

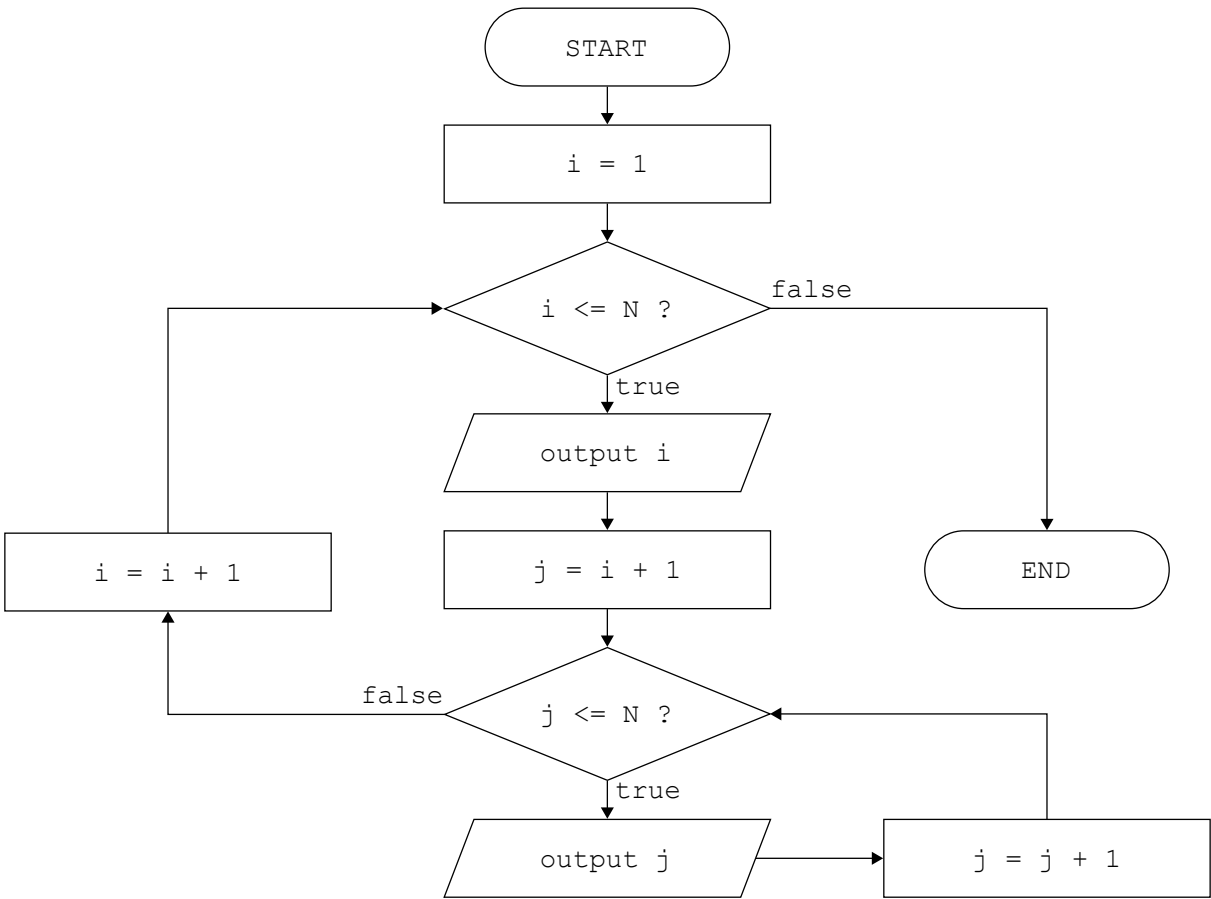
Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

1. Algorithms can be represented by flowcharts.
- (a) State **one** reason why a flowchart could be used to represent an algorithm.

[1]
- Consider the flowchart shown in **Figure 1**.

Figure 1: Flowchart for an algorithm



- (b) Copy and complete the trace table formed by the processing of this flowchart, where $N = 3$.
- [3]

i	i <= N?	j	j <= N ?	output
1				

(Option A continues on the following page)

— 3 —

(Option A, question 1 continued)

Big O notation is used to indicate the time complexity of an algorithm.

- (c) State the Big O notation for the algorithm shown in **Figure 1**. [1]

The text file, employees, contains the following information about each employee in a company:

- Surname
- First name
- Role

Each item of data is stored in a new line. Details of the first three employees are shown:

```
Roberts
John
Sales
Saied
Abdullah
Manager
Sanchez
Pedro
IT manager
...
```

An algorithm reads the data from the file, employees, line by line. The algorithm writes only the surname of each employee into a new text file, surnames. Both files are currently closed.

- (d) Describe the steps that would be followed by this algorithm. [5]

(Option A continues on page 5)**Turn over**



Blank page



(Option A, question 1 continued)

Stacks are a data structure frequently used in computer science to represent real-world scenarios.

Table 1 shows the fundamental operations that operate on a stack.

Table 1: Fundamental operations that operate on a stack

<code>push()</code>	Inserts an element onto the top of the stack.
<code>pop()</code>	Removes an element from the top of the stack.
<code>peek()</code>	Returns the top element of the stack without removing it from the stack.
<code>isEmpty()</code>	Returns true if the stack is empty, else it returns false.

(e) Sketch the stack resulting from the following operations.

```
push(1)
push(2)
push(9)
peek()
pop()
push(8)
push(10)
pop()
push(6)
```

You should clearly indicate the top of the stack. [2]

A new stack has been created and filled with elements.

(f) Using only the stack operations shown in **Table 1**, describe the steps that will locate **and** print only the **first** element that was pushed onto the new stack. [3]

(Option A continues on the following page)



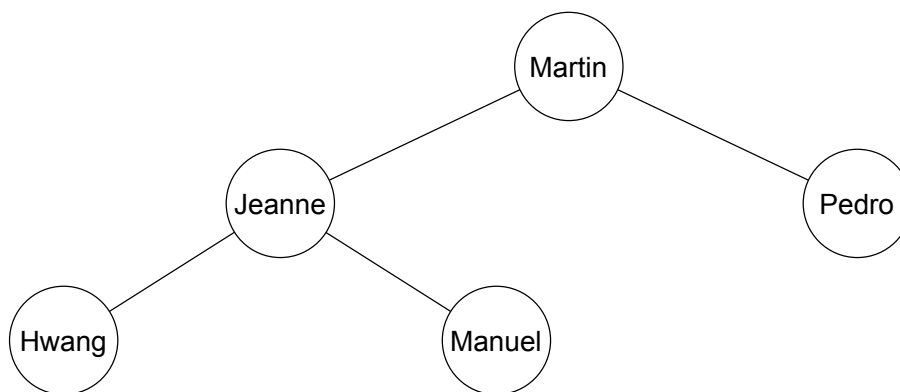
(Option A continued)

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

5. An array of names is being added to a binary search tree using a recursive insertion method. The tree, after the insertion of the first five names, is shown in **Figure 2**.

Figure 2: The binary search tree after insertion of the first five names



- (a) Sketch the binary search tree after the names “Louis” and “Indie” are inserted into it. [2]
- (b) Explain why the process of inserting a name into the binary search tree is recursive. [3]

The iterative method `iter(int x)` is shown:

```

public int iter(int x)
{
    s = 0;
    while (x > 0)
    {
        s = s + x;
        x = x - 1;
    }
    return s;
}
  
```

- (c) (i) State the value returned by the call `int result = iter(6);`. [1]
- (ii) Construct the recursive method `recur(int x)` that performs the same function as the `iter(int x)` method. [4]
- (iii) Explain why a run time error might occur in `recur()` for large values of `x`. [3]

(Option A continues on the following page)

(Option A continued)

Option B — Python

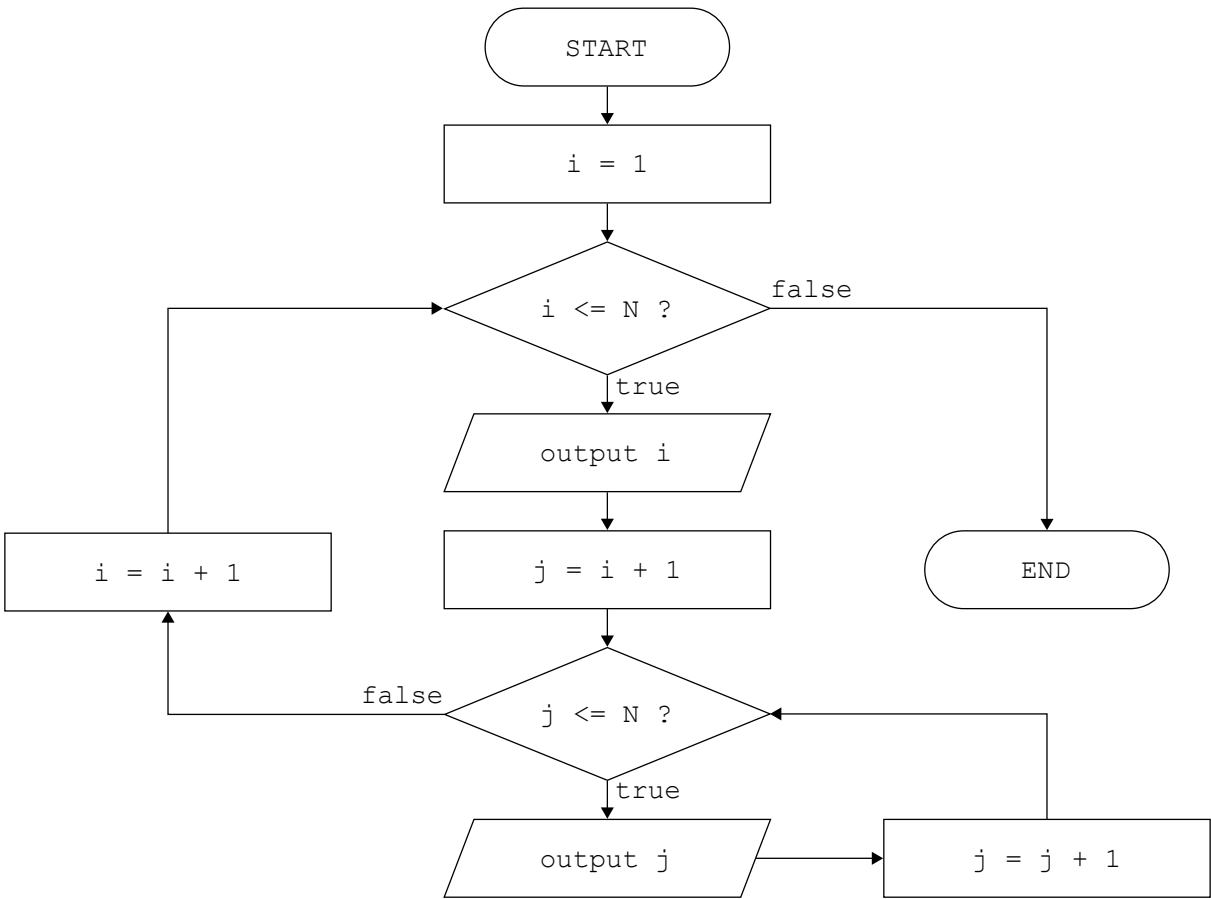
Answer **all** questions. All code that forms part of your responses must be written in Python.

7. Algorithms can be represented by flowcharts.
- (a) State **one** reason why a flowchart could be used to represent an algorithm.

[1]

Consider the flowchart shown in **Figure 3**.

Figure 3: Flowchart for an algorithm



- (b) Copy and complete the trace table formed by the processing of this flowchart, where $N = 3$.
- [3]

i	i <= N?	j	j <= N ?	output
1				

(Option B continues on the following page)

(Option B, question 7 continued)

Big O notation is used to indicate the time complexity of an algorithm.

(c) State the Big O notation for the algorithm shown in **Figure 3**.

[1]

The text file, employees, contains the following information about each employee in a company:

- Surname
- First name
- Role

Each item of data is stored in a new line. Details of the first three employees are shown:

```
Roberts
John
Sales
Saied
Abdullah
Manager
Sanchez
Pedro
IT manager
...
```

An algorithm reads the data from the file, employees, line by line. The algorithm writes only the surname of each employee into a new text file, surnames. Both files are currently closed.

(d) Describe the steps that would be followed by this algorithm.

[5]

(Option B continues on page 17)

Turn over

Blank page

— 11 —

(Option B, question 7 continued)

Stacks are a data structure frequently used in computer science to represent real-world scenarios.

Table 4 shows the fundamental operations that operate on a stack.

Table 4: Fundamental operations that operate on a stack

<code>push()</code>	Inserts an element onto the top of the stack.
<code>pop()</code>	Removes an element from the top of the stack.
<code>peek()</code>	Returns the top element of the stack without removing it from the stack.
<code>isEmpty()</code>	Returns true if the stack is empty, else it returns false.

(e) Sketch the stack resulting from the following operations.

```
push(1)
push(2)
push(9)
peek()
pop()
push(8)
push(10)
pop()
push(6)
```

You should clearly indicate the top of the stack. [2]

A new stack has been created and filled with elements.

(f) Using only the stack operations shown in **Table 4**, describe the steps that will locate **and** print only the **first** element that was pushed onto the new stack. [3]

(Option B continues on the following page)

Turn over

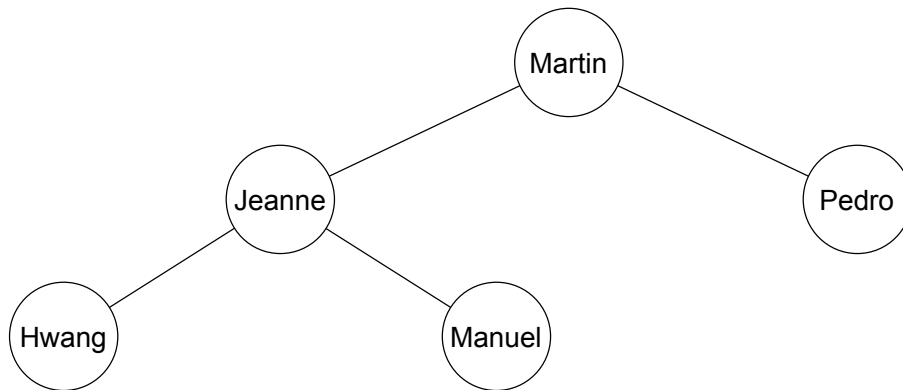
(Option B continued)

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

11. A list of names is being added to a binary search tree using a recursive insertion method. The tree, after the insertion of the first five names, is shown in **Figure 4**.

Figure 4: The binary search tree after insertion of the first five names



- (a) Sketch the binary search tree after the names “Louis” and “Indie” are inserted into it. [2]
- (b) Explain why the process of inserting a name into the binary search tree is recursive. [3]

The iterative method `iter(x)` is shown:

```
def iter(x):
    s = 0
    while (x > 0):
        s = s + x
        x = x - 1
    return s
```

- (c) (i) State the value returned by the call `result = iter(6)`. [1]
- (ii) Construct the recursive method `recur(x)` that performs the same function as the `iter(x)` method. [4]
- (iii) Explain why a run time error might occur in `recur()` for large values of `x`. [3]

(Option B continues on the following page)

(Option B continued)

APOLLO

B2 Programming

2 questions

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

2. A theatre stores the names and email addresses of its clients in two separate one-dimensional (1D) arrays, where each client's name and email address have corresponding indices. The two arrays are shown in **Table 2**.

Table 2: Name and email arrays

Name array		Email array	
Index	Name	Index	Email address
0	Maria Gonzalez	0	magonzalez24@example.net
1	Olga Sarkozky	1	osarkozky12@one2seven.org
2	Julia Brown	2	jbrown23@example.com
3	...	3	...

- (a) Explain why arrays have been chosen to hold this data instead of separate variables. [3]

A method has been constructed that will change each name so that the surname appears before the first name. For example, “Maria Gonzalez” will become “Gonzalez Maria”.

- (b) Describe the steps involved in this method.

Assume that there is only one space between each first name and surname. [4]

Consider the following two-dimensional (2D) array, `MAT`.

```

int[][] MAT = {{2, 7, 1}, {8, 9 ,0}}

```

- (c) State the code that will identify the data item 0 in `MAT`. [1]

(Option A continues on the following page)

— , —

(Option A, question 2 continued)

The following code operates on `MAT`:

```
int[][] MAT = {{2, 7, 1}, {8, 9, 0}};
int a = 0;
int b = 0;
for (int row = 0; row < 2; row++)
{
    for (int col = 0; col < 3; col++)
    {
        a = a + 1;
        int item = MAT[row][col];
        if (item%2 != 0)
            b = b + item;
    }
}
System.out.println ("The value of a is : " + a);
System.out.println ("The value of b is : " + b);
```

(d) State the output from this code.

[2]

The theatre employs a 2D array to indicate the status of each seat for a performance. `true` indicates that the seat has been sold for that performance. `false` indicates that it has not been sold.

The 2D array `seats` is used to represent the seating plan in the theatre. The array contains 40 rows and 30 columns.

For example, `seats[2][3] = false` shows that the seat in row 2, column 3 of the 2D array, `seats`, has not been sold.

After each performance, the sales data is analysed with a program that uses the method `least()`. This accepts the 2D array `seats` and outputs the row for which the lowest number of seats were sold.

(e) Construct the method `least()`.

Assume only one row matches this condition.

[5]

(Option A continues on the following page)

Turn over



(Option A continued)

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

8. A theatre stores the names and email addresses of its clients in two separate one-dimensional (1D) lists, where each client’s name and email address have corresponding indices. The two lists are shown in **Table 5**.

Table 5: Name and email lists

Name list		Email list	
Index	Name	Index	Email address
0	Maria Gonzalez	0	magonzalez24@example.net
1	Olga Sarkozky	1	osarkozky12@one2seven.org
2	Julia Brown	2	jbrown23@example.com
3	...	3	...

(a) Explain why lists have been chosen to hold this data instead of separate variables. [3]

A function has been constructed that will change each name so that the surname appears before the first name. For example, “Maria Gonzalez” will become “Gonzalez Maria”.

(b) Describe the steps involved in this function.

Assume that there is only one space between each first name and surname. [4]

Consider the following two-dimensional (2D) list, `MAT`.

```
MAT = [[2, 7, 1],
        [8, 9, 0]]
```

(c) State the code that will identify the data item 0 in `MAT`. [1]

(Option B continues on the following page)

(Option B, question 8 continued)

The following code operates on `MAT`:

```
a = 0
b = 0
for row in range (0, 2):
    for col in range (0, 3):
        a = a + 1
        item = MAT[row][col]
        if item%2 != 0:
            b = b + item
print ('The value of a is : ',a)
print ('The value of b is : ',b)
```

(d) State the output from this code.

[2]

The theatre employs a 2D list to indicate the status of each seat for a performance. `True` indicates that the seat has been sold for that performance. `False` indicates that it has not been sold.

The 2D list `seats` is used to represent the seating plan in the theatre. The list contains 40 rows and 30 columns.

For example, `seats[2][3] = False` shows that the seat in row 2, column 3 of the 2D list, `seats`, has not been sold.

After each performance, the sales data is analysed.

One program uses the function `least()`. This accepts the 2D list `seats` and outputs the row for which the lowest number of seats were sold.

(e) Construct the function `least()`.

Assume only one row matches this condition.

[5]

(Option B continues on the following page)

Turn over

(Option B continued)

APOLLO

B3 Object-oriented programming

2 questions

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

3. An airline uses object-oriented programming (OOP) to manage its booking system. This includes the class `Passenger`, which is partially shown.

```
public class Passenger{

    private String passengerName;
    private String email;
    private double ticketPrice; // the price is determined when each
                                // ticket is purchased

    public Passenger(String name, String email, double price)
    {
        this.passengerName = name;
        this.email = email;
        this.ticketPrice = price;
    }

    public String getPassengerName()
    {
        return passengerName;
    }
}
```

- (a) (i) List the operations carried out when the following code is run:

```
Passenger p1 = new Passenger("Conejo Jorge",
    "st34@one2seven.org", 240);
```

[3]

- (ii) Outline **two** consequences of the use of the modifier `private` in this class. [2]

(Option A continues on the following page)

— 3 —

(Option A, question 3 continued)

The data of each flight are stored in a `Flight` object. This includes details of the passengers stored in an array of `Passenger` objects.

The `Flight` class is partially shown:

```
public class Flight{
    public int seatsBooked;    // incremented whenever a ticket is bought
    public int maxPassengers; // maximum number of passengers for this flight
    public double initialPrice; // initial price of a ticket on this flight
    public Passenger[] passengers; // array of Passenger objects

    public Flight(int mp, double ip)
    {
        this.seatsBooked = 0;
        this.maxPassengers = mp;
        this.initialPrice = ip;
        this.passengers = new Passenger[maxPassengers];
    }

    updatedTicketPrice() // code missing

    addPassenger()      // code missing

    selectSort()        // code missing
}
```

Multiple classes have a variety of relationships, including aggregation and composition.

(b) Outline the relationship between the `Passenger` and `Flight` classes.

[2]

(Option A continues on the following page)

Turn over

(Option A, question 3 continued)

The initial price of a ticket for each flight (`initialPrice`) is the value set by the airline before any tickets have been sold for that flight. However, the actual ticket price that the passenger pays (`ticketPrice`) will change depending on the number of tickets already sold.

The airline determines the actual price of a ticket as follows:

- It is equal to the initial price if less than 10 % of tickets have been sold.
- It is 1.5 times the initial price if 10 %–50 % of tickets have been sold.
- It is 3 times the initial price if more than 50 % of tickets have been sold.

- (c) Construct the method, `updatedTicketPrice()`, that calculates and returns the actual ticket price. This method will be called before a new `Passenger` object is created. [4]

The method `addPassenger()` is called whenever a ticket is purchased by a passenger. This method is passed the name of the passenger and their email address.

This method then updates the `passengers` array and carries out any additional processing.

- (d) Construct the method `addPassenger()`. [4]

When the booking process has been completed, the `passengers` array is sorted into ascending order by passenger name. The sort routine used for this is a selection sort named `selectSort()`.

- (e) Without writing code, describe the steps carried out by the method `selectSort()`. [5]

(Option A continues on the following page)

— . . —

(Option A continued)

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

9. An airline uses object-oriented programming (OOP) to manage its booking system. This includes the class `Passenger`, which is partially shown.

```
class Passenger():

    def __init__(self, name, email, price):
        self._passenger_name = name
        self._email = email
        self._ticket_price = price # the price is determined when each
                                   # ticket is purchased

    def get_passenger_name(self):
        return self._passenger_name
```

- (a) (i) List the operations carried out when the following code is run:

```
p1 = Passenger("Conejo Jorge", "st34@one2seven.org", 240) [3]
```

The underscore character in `self._email = email` identifies the variable as being a private attribute of the class.

- (ii) Outline **two** consequences of declaring an attribute as private. [2]

(Option B continues on the following page)

— 41 —

(Option B, question 9 continued)

The data of each flight are stored in a `Flight` object. This includes details of the passengers stored in a list of `Passenger` objects.

The `Flight` class is partially shown:

```
class Flight():  
  
    def __init__(self, mp, ip):  
        self.seats_booked = 0      # incremented every time a ticket is bought  
        self.max_passengers = mp  # maximum number of passengers for this flight  
        self.initial_price = ip   # initial price of a ticket on this flight  
        self.passengers = []      # list of Passenger objects  
  
    def updated_ticket_price() # code missing  
  
    def add_passenger() # code missing  
  
    def select_sort() # code missing
```

Multiple classes have a variety of relationships, including aggregation and composition.

(b) Outline the relationship between the `Passenger` and `Flight` classes.

[2]

(Option B continues on the following page)

Turn over

(Option B, question 9 continued)

The initial price of a ticket for each flight (`initial_price`) is the value set by the airline before any tickets have been sold for that flight. However, the actual ticket price that the passenger pays (`ticket_price`) will change depending on the number of tickets already sold.

The airline determines the actual price of a ticket as follows:

- It is equal to the initial price if less than 10 % of tickets have been sold.
- It is 1.5 times the initial price if 10 %–50 % of tickets have been sold.
- It is 3 times the initial price if more than 50 % of tickets have been sold.

- (c) Construct the function, `updated_ticket_price()`, that calculates and returns the actual ticket price. This function will be called before a new `Passenger` object is created.

[4]

The method `add_passenger()` is called whenever a ticket is purchased by a passenger. This method is passed the name of the passenger and their email address.

This method then updates the `passengers` list and carries out any additional processing.

- (d) Construct the method `add_passenger()`.

[4]

When the booking process has been completed, the `passengers` list is sorted into ascending order by passenger name. The sort routine used for this is a selection sort named `select_sort()`.

- (e) Without writing code, describe the steps carried out by the method `select_sort()`.

[5]**(Option B continues on the following page)**

— 20 —

(Option B continued)

APOLLO

B4 Abstract data types-HL only

4 questions

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

4. Airport codes are used in place of airport names when displaying flight information. Some examples are shown in **Table 3**.

Table 3: Examples of airport codes

Airport code	Airport name
LHR	London Heathrow
SIN	Singapore
LAX	Los Angeles

- (a) Describe how a HashMap could be set up and used to return the airport name when an airport code is passed to the HashMap as a parameter. [3]

An airline stores the email addresses of the passengers in a separate array for each flight. It uses these arrays to send out promotional emails. Many passengers will have booked more than one flight with this airline.

- (b) Outline how a HashSet can be used to ensure that only one email is sent out to each email address. [2]

Two HashSets, `set1` and `set2`, have been created:

```
set1: [3, 6, 2, 9, 10]
set2: [2, 6, 1, 11]
```

- (c) State the output after each of the following coding statements is run:

(i) `System.out.println(set1.contains(1));` [1]

(ii) `set1.retainAll(set2);`
`System.out.println(set1);` [1]

(Option A continues on the following page)

Turn over

(Option A continued)

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

6. A computer game stores the name and score of each player in the nodes of a linked list.

The linked list makes use of the `Node` class, in which the data for each node consists of the player's name and score. This class is shown:

```
class Node{

    public String name;
    public int score;
    public Node next;

    public Node(String n, int s)
        this.name = n;
        this.score = s;
        this.next = null; // initialize next pointer as null
}
```

The `Node` object `head` will point to the first node in the linked list.

The driver class contains the method `findPlayer()`.

This method is passed the name of a player being searched for in the linked list. It will print the player's score if the name of the player is found, or a suitable message if not found.

- (a) Construct the method `findPlayer()`. [5]

The driver class will permanently maintain the nodes in the linked list in descending order of score without ever needing to sort them.

The method `addInOrder()` will accept a new node as a parameter. This node, which contains the player's data, will be inserted into the correct position in the linked list.

- (b) Without writing code, describe the steps in the method `addInOrder()`.

Assume the list is not empty. [5]

End of Option A

Turn over

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

- 10.** Airport codes are used in place of airport names when displaying flight information. Some examples are shown in **Table 6**.

Table 6: Examples of airport codes

Airport code	Airport name
LHR	London Heathrow
SIN	Singapore
LAX	Los Angeles

- (a) Describe how a dictionary could be set up and used to return the airport name when an airport code is passed to the dictionary as a parameter. [3]

An airline stores the email addresses of the passengers in a separate list for each flight. It uses these lists to send out promotional emails. Many passengers will have booked more than one flight with this airline.

- (b) Outline how a set can be used to ensure that only one email is sent out to each email address. [2]

Two sets, `set1` and `set2`, have been created:

```
set1 = {3, 6, 2, 9, 10}
set2 = {2, 6, 1, 11}
```

- (c) State the output after each of the following coding statements is run:

- (i) `print(12 in set1).` [1]
- (ii) `print(set1 & set2).` [1]

(Option B continues on the following page)

Turn over

— 47 —

(Option B continued)

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

- 12.** A computer game stores the name and score of each player in the nodes of a linked list.

The linked list makes use of the `Node` class, in which the data for each node consists of the player's name and score. This class is shown:

```
class Node():

    def __init__(self, name, score):
        self.name = name
        self.score = score
        self.next = None      # initialize next pointer as null
```

The `Node` object `self.head` will point to the first node in the linked list.

The driver code contains the method `def find_player(self, name)`.

This method is passed the name of a player being searched for in the linked list. It will print the player's score if the name of the player is found, or a suitable message if not found.

- (a) Construct the method `def find_player(self, name)`. [5]

The driver code will permanently maintain the nodes in the linked list in descending order of score without ever needing to sort them.

The method `def add_in_order(self, new_node)` will accept a new node as a parameter. This node, which contains the player's data, will be inserted into the correct position in the linked list.

- (b) Without writing code, describe the steps in the method `def add_in_order(self, new_node) :`

Assume the list is not empty. [5]

End of Option B

