

APOLLO

Computer Science SL Answers

First assessment 2027

Whole questions arranged by syllabus topic.

Original question and scheme crops preserve source diagrams.

APOLLO

A1 Computer fundamentals

1 questions

1. (a) (i) *Award [1] max*

Address (bus).

(ii) *Award [2] max*

To store intermediate arithmetic and logic results/hold operand data;
...to reduce the number of memory accesses (by holding intermediate results);

Acts as a temporary storage location;
...which stores the values and results from arithmetic and logical operations;

Holds data or the results of an operation;
...during the processing cycles;

(b) *Award [1] max*

10101100

(c) *Award [2] max*

Award [1] for each correct column (Difference, Borrower)

X	Y	Difference	Borrower
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(d) *Award [3] max*

Award up to [1] per difference

(Permanence) ROM is read-only and non-volatile, whereas flash data can be electronically erased and reprogrammed;

(Security) ROM is more secure (less corruptible) than flash due to its non-volatility;

(Longevity) ROM doesn't wear from read/write whereas flash has a limited lifespan;

(Size/capacity) Flash could be larger capacity than ROM;

(Update) Flash can be updated for user preferences whereas ROM needs CMOS RAM to store.

– 6 –

(e) *Award [2] max*

Award [1] for definition, and [1] for expansion

Processes are divided into different queues;
...based on specific criteria, e.g., separate queues for system processes, interactive user processes, background processes;

Processes are divided into different queues;
...(to allow the operating system) to allocate resources more efficiently;

Processes are divided into different queues;
...to ensure higher priority processes receive more immediate attention;

Processes are divided into different queues;
...to balance the load across different types of processes/ensure no single process monopolizes the CPU;

APOLLO

A2 Networks

2 questions

2. (a) *Award [3] max*

Award [1] per point

Transmission speed/higher bandwidth;
Lower latency compared to twisted-pair cables;
Transmits signals over greater distances than twisted-pair cables;
Immune to electromagnetic interference (EMI) and radio-frequency interference (RFI);
Better security against unauthorized access or data tapping;
Better scalability as data transmission needs grow;
Generally more durable than twisted-pair cables;
Longer lifespan and lower maintenance requirements;

(b) *Award [4] max*

Award [1] per line

The original data is divided into smaller units called packets;
Each packet is labeled (header/source and destination IP addresses, etc.);
Packets are sent over the network individually, taking different routes;
Error checking is performed at each switch/router;
At destination, they are reassembled using sequence numbers in the packet headers;
The recipient system acknowledges receipt/requests missing packets to be resent.

(c) *Award [4] max*

Award [1] for identifying a point and [1] for expansion for any two points

Broadcasts: Traffic is confined to smaller subnets/few devices on the subnet;
Devices receive fewer broadcast messages freeing up bandwidth;

Specialist transmissions: By subnetting segments traffic based on departments, the type of data traffic is separated out, avoiding data collisions;

Network stability: Less competition for resources;
To ensure the network is more stable (with fewer problems to affect congestion);

Isolation of problems: Packet collisions and broadcast storms can be isolated within a subnet, preventing them from affecting the entire network;

Prioritizing: QoS strategies to prioritize certain types of traffic;
Ensuring critical transmissions are unaffected by less critical ones.

Optimized performance: Reduced congestion;
Allowing the network to perform at an optimized level.

5. (a) *Award [2] max*

Award [1] for identifying issue and [1] for expansion

Reputational damage;

Clients may lose confidence in a company that has a breach of security;

Clients may use a different hospital/refuse to use that hospital;

Financial fraud for patients;

Patients' credit card numbers/banking data/passwords may result in funds being removed;

Theft of data;

Patients' data is used unethically (e.g., insurance company refusing to insure patients with potential pre-existing medical conditions);

Identity theft;

Having your identity stolen can cause psychological damage; e.g., a person may steal a person's online profile and insult others causing emotional distress;

Data loss causing disruption to hospital services;

Phishing attacks are often designed to steal sensitive data; e.g. doctors lose access to patients records and can't treat them;

Legal consequences;

Company may be sued for leaking sensitive information by patients/companies/government/ombudsman/lose license to practice medicine;

Create vulnerability for additional malware;

Causing further disruption.

(b) *Award [4] max*

Award [1] for identifying method and [1] for expansion

Packet filtering: a firewall inspects incoming data packets;

Blocking or allowing packets based on predefined rules (e.g., content, location sent from);

Stateful inspection: Monitors active connections;

Only packets that come from an established connection or session are accepted;

Proxy service: Firewalls can act as a proxy inspecting and forwarding packets;

Prevents direct access to a system,

Blocking unwanted traffic: Preventing access from harmful sites;

Configured to block data from certain locations or applications;

Network address translation (NAT): Presents a single, public-facing IP address;

Making it harder for attackers to find individual systems;

Application layer filtering: Inspect data up to the application layer;

Blocking specific activities within protocols, like file uploads, instant messaging;

Intrusion detection and prevention: Mechanisms to identify suspicious patterns;

Alerting administrators or actively blocking the suspicious traffic;

– 11 –

Segmentation: Firewalls can be used to segment different parts of the network;
Sensitive data/operations can be kept separate from general use;

Virus and malware scanning: integrated malware scanners;
Detecting malware as soon as it enters the network;

Denial of service protection: limit the rate of requests from a particular IP address;
Helps to mitigate distributed denial-of-service (DDoS) attacks;

User authentication: can be set up to require user authentication;
So that only registered users can access the network resources;

Mark [2] + [2]

(c) **Award [6] max**

Responses may include some of the following points. The best ones will demonstrate research and give a balanced perspective before presenting a conclusion.

Steps to reduce potential risks

- Implement robust security protocols: TaylorMade Healthcare can establish the following steps.
- Security protocols: Ensure there is a failover system for firewalls, security software, email filtering, and regular software updates.
- Device management: Implement a mobile device management (MDM) solution to control which devices can access company data, enforce security policies, and remotely wipe data if necessary.
- VPN and secure connections: Ensure that all devices connecting to the company network do so through a secure connection like a VPN.
- Regular audits: Periodically check and verify the security status of all devices that connect to the network.
- Data segregation: Use applications that can segregate corporate data from personal data on the device.
- User awareness training: Educating staff about potential threats, such as phishing scams and ransomware attacks, can prevent many attacks. Candidates should be aware that human error remains a significant challenge. TaylorMade Healthcare should have clearly defined procedures and protocols for new staff and departing staff, such as not connecting to open Wi-Fi networks and not downloading untrusted apps.
- Employees should be made aware of the damage caused by malware.

Risks for bring your own device (BYOD) policy

- Device diversity: A variety of device types, operating systems, and software versions can make standardized security measures challenging.
- Uncontrolled environments: Personal devices might connect to insecure networks, increasing the risk of exposure to threats.
- Data leakage: There is a risk of sensitive organizational data being stored alongside personal data and potentially being shared inadvertently.
- Loss or theft: Personal devices can easily be lost or stolen, risking the exposure of any stored organizational data.

– 12 –

Ransomware

- A type of malware that infiltrates and encrypts data, holding it hostage until a ransom is paid.
- Typically infiltrates through phishing emails, drive-by download, malvertising, exploit kits, removable media, and remote desktop protocols (RDP) exploits.
- Once activated, the ransomware payload runs, sometimes with a delay or when triggered by specific user actions.
- The ransomware scans the network for specific file types and begins to encrypt them. The encryption keys are typically held only by the attacker.
- After encryption, the ransomware communicates with a command and control (C&C) server operated by the attacker, signalling the attack has been successful.
- The victim is presented with a time-dependent demand for payment, typically in cryptocurrency to maintain the attacker's anonymity.
- If the victim pays, the attacker may send the decryption tool or key.
- Some ransomware strains delete themselves and activities after the attack, making it harder for authorities to trace them.

Rootkit attacks:

- A stealthy type of malware designed to remain hidden, allowing attackers to maintain prolonged unauthorized access.
- Typically infiltrates systems through exploiting vulnerabilities, bundled with other malware, or malicious software installations.
- Once installed, the rootkit may modify system binaries or install itself as a kernel module or driver.
- The rootkit malware can intercept system calls and manipulate output to hide its presence.
- Communicates with a command and control (C&C) server, allowing the attacker to remotely execute commands and steal data.
- Advanced rootkits can patch system updates or alter firmware, persisting even if the operating system is reinstalled.
- Many rootkits include functions to download and install additional malware, transforming the compromised machine into a bot.
- Signature-based approaches often fail to identify rootkits so behavioural-based detection is required to identify and remove them.

Conclusion

A reasoned conclusion that expresses an opinion and is supported by evidence.

Marks	Level descriptor
No marks	• No knowledge or understanding of the relevant issues and concepts. • No use of appropriate terminology.
Basic 1–2 marks	• Minimal knowledge and understanding of the relevant issues. • Minimal use of appropriate terminology. • The answer may be little more than a list. • No reference is made to the case study or independent research.
Adequate 3–4 marks	• A descriptive response with limited knowledge and/or understanding. • A limited use of appropriate terminology. • There is some analysis. • There is evidence of some research.
Competent 5–6 marks	• Knowledge and understanding of the related issues and/or concepts. • Uses terminology appropriately in places. • There is evidence of analysis. • There is evidence of research. • There is conclusion

APOLLO

A3 Databases

1 questions

3. (a) *Award [2] max*

Award [1] for stating a benefit and [1] for expansion

Provides data integrity;
Accurate and reliable data/primary and foreign keys prevent orphan data/enforce relationships;

Maintains data consistency;
Ensures data in one part of the database does not contradict data in another;

Allows data concurrency;
Multiple users can access data at the same time;

Reduces of data redundancy;
Uses normalization to ensure data is only stored once;

Allows for complex querying;
Uses a declarative/industry standard language;

(b) *Award [1] per point, up to [2] max*

Data volume growth (leading to performance issues);
Increased concurrency (an increase in users may lead to concurrency issues);
Complexity of queries (more sophisticated analysis use more resources);
Hardware limitations (single server becomes overloaded);
Load balancing on multiple servers;
Inefficient indexing (affecting time to search);

(c) *Award [3] max*

Award [1] for SELECT DISTINCT or GROUP BY

Award [1] for using LIKE

Award [1] for using the wildcard (any wildcard accepted) and therapy

```
SELECT DISTINCT PatientName FROM VISITS  
WHERE Treatment LIKE '%therapy%';
```

Alternative solution using group by:

```
SELECT PatientName FROM VISITS  
WHERE Treatment LIKE '%therapy%'  
GROUP BY PatientName
```

Alternative solution using a subquery and group by:

```
SELECT PatientName  
FROM (  
    SELECT PatientName  
    FROM VISITS  
    WHERE Treatment LIKE '%therapy%'  
) AS subquery  
GROUP BY PatientName;
```

– 9 –

(d) *Award [5] max*

Award [1] for the PATIENT table with correct Primary Key

Award [1] for the DOCTOR table with correct Primary Key

Award [1] for PatID foreign key in the VISIT table

Award [1] for DocID foreign key in the VISIT table

Award [1] for VisitID or PatID/DocID/VisitDate composite key in the VISIT table

```
PATIENT(PatID, PatientName, PatientDOB)
DOCTOR(DocID, Doctor)
VISIT(VisitID, PatID, DocID, VistDate, Treatment)
```

APOLLO

A4 Machine learning

1 questions

4. (a) Award [2] max

Award [1] per point

Dataset size/data volume/number of features;
Algorithm/model complexity;
Feature engineering;
Hyperparameter tuning;
Real-time requirements;
Ensemble methods.

(b) Award [2] max

Award [1] per point, from any section

Audit: Carry out independent audits to check for algorithmic bias;
Validate: Run validations to ensure that the algorithm is not biased;
Cross-reference diagnosis with human diagnosis;

Public disclosure: Inform patients about the algorithm and its limitations;
Provide informed consent forms for patients to sign;

Committee: Set up an oversight committee to approve the use of machine learning;
Any significant change to algorithms or datasets requires committee approval;
Have a patient representative on the committee;

– 10 –

Section B

APOLLO

B1 Computational thinking

2 questions

1. (a) *Award [1 max]*

because it is in diagrammatic form;
 it is program agnostic;
 no complicated code / syntax to follow;

(b) *Award [3]*

correct values of i;
correct values of j;
correct output;

i	i <= N ?	j	j <= N ?	output
1	TRUE			1
		2	TRUE	2
		3	TRUE	3
		4	FALSE	
2	TRUE			2
		3	TRUE	3
		4	FALSE	
3	TRUE	4	FALSE	3
4	FALSE			

(c) *Award [1]. Note: the algorithm shows a nested loop.*

$O(n^2)$;

– 6 –

(d) Award **[5 max]** as follows

opening *employees* file for reading / default mode;
open *surnames* file for writing;
looping through *employees* file;
use of a counter or equivalent;
writing surnames only to *surnames* file;
closing both files;

*Note: Award **[1]** for first 2 marking points if files are opened without indicating access modes.*

Algorithm is:

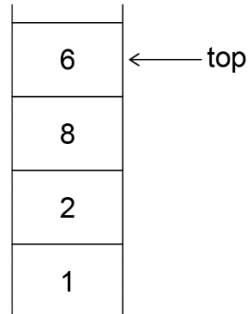
```
open employees file for reading / in default mode
open surnames file for writing
set counter to 0
repeat until end of employees file
    read next line from employees file
    if counter%3 = 0 *
        write line to surnames file
    increment counter
close both files
```

* there are alternatives ways of using counter

– 7 –

(e) *Award [2].*

correct elements in stack **[1]**;
 correct indication of top **[1]**;

(f) *Award [3] as follows:*

Award [1] for each correct line (or equivalent).

repeat until isEmpty is true / while isEmpty is false;

pop an element and save its value/save in a variable/push it onto a new stack (stack2)

print the last value saved/print the variable/pop stack2 and print value

Note: Answer must demonstrate an algorithmic approach.

4. (a) Award [1 max]

because it is in diagrammatic form;
 it is program agnostic;
 no complicated code / syntax to follow;

(b) Award [3]

correct values of i;
correct values j;
correct output;

i	i <= N ?	j	j <= N ?	output
1	TRUE			1
		2	TRUE	2
		3	TRUE	3
		4	FALSE	
2	TRUE			2
		3	TRUE	3
		4	FALSE	
3	TRUE	4	FALSE	3
4	FALSE			

(c) Award [1]. Note: the algorithm shows a nested loop.

$O(n^2)$;

– 13 –

(d) Award **[5 max]** as follows

opening *employees* file for reading / default mode;
open *surnames* file for writing;
looping through *employees* file;
use of a counter or equivalent;
writing surnames only to *surnames* file;
closing both files;

*Note: Award **[1]** for first 2 marking points if files are opened without indicating access modes*

Algorithm is:

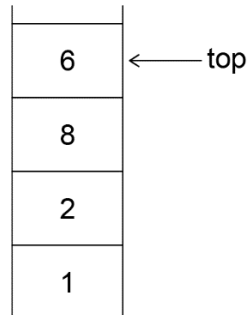
```
open employees file for reading / in default mode
open surnames file for writing
set counter to 0
repeat until end of employees file
  read next line from employees file
  if counter%3 = 0 *
    write line to surnames file
  increment counter
close both files
```

* there are alternatives ways of using counter

– 14 –

(e) *Award [2].*

correct elements in stack **[1]**;
correct indication of top **[1]**;



(f) *Award [3] as follows:*

Award [1] for each correct line (or equivalent).

```
repeat until isEmpty is true / while isEmpty is false;  
  pop an element and save its value/save in a variable/push it onto a new stack (stack2);  
  print the last value saved/print the variable/pop stack2 and print value;
```

Note: Answer must demonstrate an algorithmic approach.

APOLLO

B2 Programming

2 questions

2. (a) Award [3 max]

It is easier to work with one single structure than several individual ones;
Individual elements that have the same value can be quickly accessed via their indices;
Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) Award [4 max]

repeat for each entry/index in the `Name` array
 identify the position of the space in each name
 save the substring to the left as `s1` and the substring to the right as `s2`
 save the new entry as `s2+space+s1`

(c) Award [1]

```
MAT[1][2];
```

(d) Award [1] each;

The value of `a` is: 6
The value of `b` is: 17

– 9 –

(e) *Award [5 max] as follows:**Award [1] for initial variables correctly set;**Award [1] for both loops;**Award [1] for checking seat for False;**Award [1] for updating row value;**Award [1] for resetting tempTotal;**Award [1] for printing the correct row;*

```

public void least(boolean[][]seats)
{
    int falseTotal = 0;
    int leastRow = 0;

    for (int x = 0; x < 40; x++)
    {
        int tempTotal = 0;
        for (int y = 0; y < 30; y++)
        {
            if (seats[x][y] == false)           // if(!seats[x][y] )
                tempTotal = tempTotal + 1;
        }
        if (tempTotal > falseTotal)
        {
            leastRow = x;
            falseTotal = tempTotal;
        }
    }
    System.out.println ("Row with least seats sold is " + leastRow);
}

```

5. (a) Award [3 max]

It is easier to work with one single structure than several individual ones;
 Individual elements that have the same value can be quickly accessed via their —indices;
 Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) Award [4 max] as follows:

repeat for each entry/index in the `Name` list
 identify the position of the space in each name
 save the substring to the left as `s1` and the substring to the right as `s2`
 save the new entry as `s2+space+s1`

repeat for each entry/index in the `Name` list
 use the `split()` function
 to separate the 2 strings into 2 list values (`s1` and `s2`)
 save the new entry as `s2+space+s1`

(c) Award [1]

```
MAT[1][2];
```

(d) Award [1] for each:

The value of `a` is: 6
 The value of `b` is: 17

(e) Award [5 max] as follows:

Award [1] for initial variables correctly set;
Award [1] for both loops;
Award [1] for checking seat for `False`;
Award [1] for updating row value;
Award [1] for resetting `temp_total`;
Award [1] for printing the correct row;

```
def least(seats):
    false_total = 0
    least_row = 0

    for x in range(40):
        temp_total = 0
        for y in range(30):
            if seats[x][y] == False:
                temp_total = temp_total + 1
        if temp_total > false_total:
            least_row = x
            false_total = temp_total

    print("Row with least seats sold is ", least_row)
```

APOLLO

B3 Object-oriented programming

2 questions

3. (a) Award [4 max]

creates / assigns memory to a new instance of the `SalesPerson` class;
 identified by the variable `employee1`;
 initialises `id` with value "A6457098";
 declares (empty) sales array;

(b) Award [2 max]

`bonus1` belongs to the class / it will remain the same for all instances of the class / has the same memory for all instances;
`id` belongs to the object/instance / it will be different for each instance / be assigned different memory for each instance;

`id` is declared inside the constructor method;
`bonus1` is declared inside the class definition but outside any of the instance methods and constructor;

`id` gets created when an instance of the class is created;
`bonus1` is created when the program begins to execute;

changes made to instance variable `id` through one object will not reflect in another object;
 changes made in the class variable `bonus1` will reflect in all objects;

(c) Award [2 max]

provides greater security for the values stored in these variables/for the code;
 restricts access to such variables from outside of the class /only allows access through public / getter, setter methods;
 (helps to) prevents accidental/unintended changes to the variables (from outside of the class);

(d) Award [2 max]

Award [1] for parameters;

Award [1] for adding the sales figure to the array;

```
public void addSales(double amount,int month)
{
    sales[month] = amount;
}
```

– 11 –

(e) Award [5 max]

Award [1] for a correct loop;
Award [1] for initializing highest;
Award [1] for the comparison;
Award [1] for updating highest;
Award [1] for returning highest;

```
public double calculateHighest()
{
    double highest = 0;
    for (int x = 0; x < 12; x++)
    {
        if (this.sales[x] > highest)
            highest = this.sales[x];
    }
    return highest;
}
```

(f) Award [5 max]

Award [1] for a loop and initialising and updating bonus variable;
Award [1] for a correct comparison using `calculateTotal` method;
Award [1] for a correct comparison using `calculateHighest` method;
Award [1] for a correct identification of `bonus1` and `bonus2`;
Award [1] for prints out the id (using getter) and bonus (allow values only);

```
public void bonus(SalesPerson[] employee)
{
    for (int x = 0; x < 10; x++)
    {
        int bonus = 0;
        if (employee[x].calculateTotal() > 300000)
            bonus = bonus + SalesPerson.bonus1;
        if (employee[x].calculateHighest() > 20000)
            bonus = bonus + SalesPerson.bonus2;
        System.out.println ("ID is " + employee[x].getId());
        System.out.println ("Bonus is " + bonus);
    }
}
```

Option B — Python

6. (a) Award [4 max]

creates / assigns memory to a new instance of the `SalesPerson` class;
 identified by the variable `employee1`;
 initialises `id` with value "A6457098";
 declares (empty) sales list;

(b) Award [2 max]

`bonus1` belongs to the class / it will remain the same for all instances of the class / has the same memory for all instances;
`self.id` belongs to the object/instance / it will be different for each instance / be assigned different memory for each instance;

`self.id` is declared inside the constructor/ the `__init__()` method;
`bonus1` is declared inside the class definition but outside any of the instance methods and constructor;

`self.id` variable gets created when an instance of the class is;
`bonus1` is created when the program begins to execute;

changes made to instance variable `self.id` through one object will not reflect in another object; changes made in the class variable `bonus1` will reflect in all objects;

(c) Award [2 max]

provides greater security for the values stored in these variables/for the code;
 restricts access to such variables from outside of the class /only allows access through public / getter, setter methods;
 (helps to) prevents accidental/unintended changes to the variables (from outside of the class);

(d) Award [2 max]

Award [1] for parameters;

Award [1] for adding the sales figure to the list;

```
def add_sales(self, amount):
    self._sales.append(amount)
```

```
def add_sales(self, amount):
    self._sales = self._sales + [amount]
```

– 17 –

(e) **Award [5 max]**

Award [1] for a correct loop;
Award [1] for initializing highest;
Award [1] for the comparison;
Award [1] for updating highest;
Award [1] for returning highest;

```
def calculate_highest(self):
    highest = 0
    for x in range (12):                # ... range (0,12)
        if self._sales[x] > highest:
            highest = self._sales[x]
    return highest
```

```
def calculate_highest(self):            1
    highest = 0
    for x in range (len(self._sales)):
        if self._sales[x] > highest:
            highest = self._sales[x]
    return highest
```

```
def calculate_highest(self):
    highest = 0
    for sales_figure in self._sales:
        if sales_figure > highest:
            highest = sales_figure
    return highest
```

(f) **Award [5 max]**

Award [1] for a loop and initialising and updating bonus variable;
Award [1] for a correct comparison using `calculate_total` method;
Award [1] for a correct comparison using `calculate_highest` method;
Award [1] for a correct identification of `bonus1` and `bonus2`;
Award [1] for prints out the id (using `getter`) and bonus (allow values only);

```
def calculate_bonus():
    for x in range(10):                # ... range (0,10)    ... range
        (len(self._sales))

        bonus = 0
        if employee[x].calculate_total() > 300000:
            bonus = bonus + SalesPerson.bonus1
        if employee[x].calculate_highest() > 20000:
            bonus = bonus + SalesPerson.bonus2
        print (f"Employee {employee[x].get_id()} has bonus ", bonus).
```