2226 – 4427M

Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

May 2026

Computer science

Standard level

Paper 2

38 pages

Subject details: Computer science HL paper 2 markscheme

Mark allocation

Candidates are required to answer **all** questions in **one** Option. Total 65 marks.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

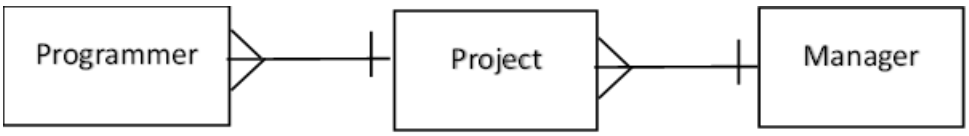
- Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).
- An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.
- Words in (…) in the markscheme are not necessary to gain the mark.
- If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.
- Mark positively. Give candidates credit for what they have achieved and for what they have got correct, rather than penalizing them for what they have not achieved or what they have got wrong.
- Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.
- Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with "**FT**".

General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	• In the case of an "identify" question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers. • In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. • In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

Option A — Databases

1. (a) *Award [2 max]*
Award [1] for the correct relationship between Programmer and Project entities.
Award [1] for the correct relationship between Project and Manager entities.



Note: Ignore any relationship between manager and programmer.
***Note**: Accept other notations for the relationships e.g., 1:M*

(b) *Award **[2]** max.*
*Award **[1]** for each point.*

Databases have high data integrity;
All changes are consistently recorded/can be retrieved;

Databases provide version control (history of code modifications);
Allowing for easy rollback to previous versions;

Provides a clear audit trail of who made the changes;
Aiding in accountability and transparency;

Automatic backup/rollback;
Code stored in the database can be restored.

Databases ensure concurrency control;
Enabling multiple programmers to work on the same project simultaneously;

Databases have security features;
So external parties cannot access the company's codes;

Note: *Do not award marks across different clusters*

(c) *Award **[2]** max.*
*Award **[1]** per point from either section.*

Ensure that transactions are processed in the correct order;
Preventing conflicts and reducing the likelihood of update anomalies;

Help in tracking the exact time each transaction occurs;
Ensuring the database reflects the accurate sequence of operations.

They are used to reconstruct the database to a specific, consistent point in time;
Allowing rollback in the correct order;

Help determine the last successful transaction;
Ensuring incomplete/corrupted/failed updates can be rolled back or redone

Note: *Do not award marks across different clusters*

(d) *Award [**5**] max.*
*Award **[1]** for identifying the method;*
*Award **[3]** for expansion of the method;*
*Award **[1]** for reasoned argument;*

O*ptimistic concurrency control(OCC)*;
Allows multiple users to access the unmodified version of data at the same time;
Uses read isolation;
... To prevent dirty reads (reading the uncommitted changes of the other programmer);
Uses write validation (on update request) / checks if the original data has been modified by the other programmer since it was initially read;
... To prevent lost update / overwriting the other programmer's update;

Multi-version concurrency control (MVCC);
Uses multiple versions of the code to allow both programmers to work simultaneously;
…This reduces the waiting caused by locks;
A new version of the code is created anytime an update occurs (and the old version is kept temporarily);
Transactions which start before the update see the old version, transactions which start after the update see the new version;
… Allowing each programmer to see a consistent snapshot of the database;

Pessimistic concurrency control (PCC)/single-phase locking/Locking;
When a code line is edited, the database locks that line;
... to prevent dirty reads (reading uncommitted changes) / lost update (overwriting the other programmer's update);
Changes to the locked edited line are committed to the database;
After committing the changes, the lock is released;
Conflicts are prevented before they occur.

Two-phase locking (2PL);
Locks are acquired incrementally (without releasing any until all needed locks are obtained);
Shared locks can be acquired for only reading code, but exclusive locks must be acquired to update code;
Changes are made and committed while holding exclusive (all required) locks;
... to prevent dirty reads / lost updates.
Locks are released only after the commit;
Conflicts are managed by preventing access to locked resources until they are released;

Distributed version control system (DVCS
Each programmer has a local copy of the database;
Updates are made to the local database;
When the update is completed, the local database is merged back into the main database;
Conflicts are flagged and resolved by the programmers;

(e) (i) *Award **[2]** max.*
*Award **[1]** per point from any section.*

A consistent database state is one where all data integrity constraints are satisfied;
For example, unique primary keys, no hanging records, and validation/domain rules are obeyed;

A consistent database state is one where the data is accurate and free from contradictions, anomalies or errors;
The data correctly represents the intended data model/real-world situation;

A consistent database state is one reached after transactions have completed correctly;
No partial/failed/uncommitted transaction has left the database corrupted;

***Note**: Do not award marks from different clusters*

(ii) *Award **[3]** max from one section only.*
*Award **[1]** per point from one section only.*

Log recovery;
When a crash occurs, the system consults the log files to determine the last committed transactions;
The log files contain records of all transactions/database changes;
The system reapplies the committed transactions from the logs to the database;
Uncommitted in-progress transactions are rolled back using the log files.

Shadow paging;
The database maintains two-page tables: the current page table and a shadow page table;
When a transaction begins, the shadow page table is a copy of the current page table;
Updates are made to the current page table while the shadow page table remains unchanged;
If a crash occurs before the transaction commits, the system reverts to the shadow page table to restore the database to its consistent state;

Checkpointing;
The system periodically creates checkpoints by saving the current state of the database and transaction logs;
Checkpoints reduce recovery time by limiting the amount of log data that needs to be processed during recovery;
During recovery, the system starts from the last checkpoint and applies only the transactions recorded after that point;
Transactions committed before the checkpoint are considered stable and do not need to be redone;

Deferred update recovery;
Transactions record their updates in a log file without altering the actual database until commit time;
No undo needed for uncommitted transactions/If a crash occurs before a transaction commits, there is no need to roll back;
Apply committed transactions upon recovery;

Immediate update recovery;
Changes made by a transaction are written to the database before the transaction is committed;
The system keeps a transaction log (containing old values and new values) to enable recovery;
Rollback (undo) uncommitted transactions and redo the committed transactions during recovery;

2. (a) *Award **[2]** max.*
Award [2] for the correct answer

ProjectID / (accept a composite key that includes ProjectID);
It is unique / the other fields could repeat;

(b) *Award **[2]** max.*
*Award **[1]** per line from any section.*

BOOLEAN;
0 for High, 1 for Low (or equivalent) / because there are only two values;

VARCHAR/TEXT/STRING;
With validation to ensure only 'High' or 'Low' entered e.g. CHECK (Priority IN ('High', 'Low')) / because the data can describe the priority/it is a combination of characters;

ENUM;
With predefined values of High and Low;

(c) *Award **[5]** max.*
Award [1] for selecting correct fields (ClientName and ContactNumber)
Award [1] for joining CLIENT and PROJECT on ClientID
Award [1] for joining PROJECT and MANAGER on ManagerID
Award [1] for WHERE ManagerName = "Sehee Choi" // Award without quotes
Award [1] for AND Priority = "High" // Award without quotes

Example 1

```
SELECT CLIENT.ClientName, CLIENT.ContactNumber
FROM CLIENT, PROJECT, MANAGER
WHERE PROJECT.ClientID = CLIENT.ClientID
     AND PROJECT.ManagerID = MANAGER.ManagerID
     AND MANAGER.ManagerName = 'Sehee Choi'
     AND PROJECT.Priority = 'High';
```

Example 2

```
SELECT CLIENT.ClientName, CLIENT.ContactNumber FROM CLIENT
INNER JOIN PROJECT ON PROJECT.ClientID = CLIENT.ClientID
INNER JOIN MANAGER ON PROJECT.ManagerID = MANAGER.ManagerID
WHERE MANAGER.ManagerName = 'Sehee Choi'
  AND PROJECT.Priority = 'High';
```

Accept answers that aren't written in SQL. For example:

Select the ClientName and ContactNumber from the CLIENT table;
Join the CLIENT table to the PROJECT table using ClientID;
Join the MANAGER table to the PROJECT table using ManagerID;
Filter the results when the ManagerName is 'Sehee Choi';
and where the 'Priority is High.

(d) *Award **[5]** max.*
Award [1] for identifying three tables with correct fields (without PK and FK)
Award [1] for PROGRAMMER table: PK and correct fields.
Award [1] for COURSE table: PK and correct fields /correct COMPANY and QUALIFICATION tables (example 3).
Award [1] for TRAINING table: PK and non-key fields (CourseDate, Complete, Cost).//ignore any other field included;
Award [1] for one foreign key in TRAINING
Award [1] for the remaining foreign key(s) in TRAINING

Example 1

PROGRAMMER(ProgrammerID, FirstName, LastName, Level)
COURSE(CourseID, Company, Qualification)
TRAINING(ProgrammerID*, CourseID*, CourseDate, Cost, Complete)

Example 2

PROGRAMMER(ProgrammerID, FirstName, LastName, Level)
COMPANY(CompanyID, CompanyName)
QUALIFICATION (QualificationID, QualificationName)
COURSE (CourseID, CompanyID*, QualificationID*)
TRAINING(ProgrammerID*, CourseID*, CourseDate, Cost, Complete)

Example 3

PROGRAMMER (ProgrammerID, FirstName, LastName, Level)
COMPANY (CompanyID, CompanyName)
QUALIFICATION (QualificationID, QualificationName)
TRAINING(ProgrammerID*, CompanyID*, QualificationID* CourseDate, Cost, Complete)

***Notes**:*

1. *Accept a surrogate key in TRAINING e.g., TrainingID*
2. *Accept ProgrammerName instead of FirstName, LastName*
3. *Accept **Cost** in COURSE table for example 1 and 2*
4. *Award the mark for FK if not explicitly indicated as long as the name matches the PK*
5. *Award marks from one example only.*
6. *The relation names do not matter; check the fields used*

3. (a) Award ***[2]*** *max.*
Award ***[1]*** *for identifying a valid role of the conceptual schema*
Award ***[1]*** *for a brief expansion/example.*

A conceptual schema provides an overall or global view of the database;
It shows the main entities, attributes and relationships in the database;

A conceptual schema describes the organisation of the data at a high level;
It hides physical storage details such as file organisation, indexes, and storage location;

A conceptual schema acts as the basis for later logical and physical database design;
It guides how tables, keys, relationships and constraints will later be developed;

Note: *Do not award marks across different clusters*

(b) Award ***[2]*** *max.*
Award ***[1]*** *per point.*

Ensures that data definitions are uniform across the database;
Promotes consistent use of terms and data types;

Provides clear documentation of the database schema/Provides metadata;
Aids understanding of the structure and relationships of the data;

Helps maintain data integrity;
By defining the properties and constraints of each data element;

Facilitates the planning, development, and maintenance of the database;
Serves as a centralised reference for all data elements.

Note: *Do not award marks across different clusters*

(c) *Award* ***[3]*** *max.*
Award ***[1]*** *per point.*

DDL is used to define the schema of the database / it allows the definition of tables, keys, and fields;
Definition of constraints and validation rules to enforce data integrity / specification of keys and creation of relationships;
Enable the modification of database schema;
Helps document the database schema, making it easier to maintain;
Automate the setup of database structures, ensuring consistency;
Award a valid DDL command term e.g., CREATE, DROP, ALTER'

(d) *Award **[3]** max.*
*Award **[1]** per point.*

Sets up access control/user accounts;
Assign roles, permissions and access rights;
Regularly monitors database activity/reviews audits logs;
Applies encryption to sensitive data;
Applies security patches and updates;
Backup and recovery planning;
Provides guidance on secure database access;

(e) *Award [**5**] max.*
*Award **[4]** marks from 2 clusters only;*
*Award **[1]** for reasoned argument;*

It ensures compliance and legal protection;
Regulators can verify that the company is following labour laws (e.g., maximum working hours or mandatory break times);
This can reduce/prevent fines due to non-compliance;

Improves stakeholder/public trust/confidence in the company;
Allowing regulator access shows that the company is transparent/open and accountable;
Law enforcement officers are less likely to initiate disruptive audits/investigations;

It enhances the privacy of employee data;
Regulators can ensure the company is handling employee personal data according to frameworks like GDPR;
Which enhances the company's reputation;

It supports accurate reporting;
Regulators can verify records such as employee hours, contracts, tax details, or working conditions / Provides guidance for the company/help identify issues;
Thus ensuring the data is correct;

Regulator access motivates the company to maintain strong data access restrictions;
Regulators can also help identify (security) issues with the database;
And suggest strategies for improving/solving such issues;
Resulting in improved data security for the company;

Accept other reasonable expansion points.

Option B — Modelling and simulation

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

50 (%)

(ii) *Award **[2]** max.*
*Award **[1]** for seeing* `B2/40` *and* `C2/60`*;*
*Award **[1]** for correctly weighted sum;*

Example answers:
```
=  B2/40*25 + C2/60*75            // accept with any correct extra brackets
= (0.25*B2/40 + 0.75*C2/60)*100
=  B2/1.6 + B2/0.8
```

*Award a maximum of **[1]** for an unweighted mean or forgetting to change to %;*
```
= (B2/0.4 + C2/0.6)/2
```

(b) *Award **[3]** max.*
Answer using `VLOOKUP`
*Award **[1]** for using* `VLOOKUP` *on D2;* *// do not accept* `LOOKUP`
*Award **[1]** for the correct range (with or without* `$`*);*
*Award **[1]** for stating column 2;* *//* `TRUE` *is optional*

Example answers:
```
= VLOOKUP(D2, $H$2:$I$6, 2, TRUE)
= VLOOKUP(D2, H2:I6, 2)
```

Alternative answer using nested `IF`
*Award **[1]** for* using nested `IF`*;*
*Award **[1]** for correctly calculating one grade;*
*Award **[1]** for correctly calculating all grades;*

Example answers:
```
= IF(D2>=85, "A",
     IF(D2>=70, "B",
          IF(D2>=60, "C",
               IF(D2>=47, "D",
                    IF(D2>=35, "E", "")))))

= IF(D2<35, ”",
     IF(D2<47, ”E”,
          IF(D2<60, ”C”,
               IF(D2<70, ”B”,
                    IF(D2<85, ”B”, ”A”)))))
```

(c) *Award **[2]** max.*
Award [1] for not finding (the location of) a space;
Award [1] for identifying that no integer returns an error;

Example answer:
Because there is no space in A4, `FIND` cannot return an integer and an error is returned.

(d) *Award **[3]** max.*
*Award **[1]** for using an IF statement;*
*Award **[1]** for returning the space location i.e.* FIND(" ", A4);
*Award **[1]** for returning FALSE if there is no space;*

Example answers for full marks:

```
= IF(ISNUMBER(FIND(A2," ")), FIND( A2," "), FALSE)
= IFERROR(FIND( A2," "), FALSE)
= IF(ISERROR(FIND(A2," ")), FALSE, FIND( A2," "))
```

Accept alternative answers if the same result is achieved.
Beware of circular definitions that define F2 in terms of F2.

(e) **Method 1 – remove the first name and try to find a space in the remainder**
*Award **[4]** max.*
*Award **[1]** for placing middle and surname in a new column / removing first name;*
*Award **[1]** for checking if there is a second space (using an IF statement);*
*Award **[1]** for handling a potential error (with ISNUMERIC or ISERROR);*
*Award **[1]** for extracting the last name (using MID) regardless of whether there is one name or two names;*

Example answers:
Use LEFT or MID to remove the first (and place remaining text in an empty column).
Use an IF statement to check if there is another space (in the remaining text).
Use ISNUMERIC or ISERROR to handle a potential error.
Use MID or RIGHT to extract the last name.

For example, possible middle names and surnames have been placed in cell D4:

```
= IF(F2 = ISNUMBER(FIND(D4," ")),
                   MID(A2,FIND( A2," ")+1,LEN(A2)),
                   MID(MID(A2,FIND( A2," ")+1,LEN(A2)),
                       FIND(" ",MID(A2,FIND( A2," ")+1,LEN(A2)))+1,
                       LEN(D2)))
```

Method 2 – try to find two spaces
*Award **[4]** max.*
*Award **[2]** for using IFERROR() to return two answers (equivalent to IF and ISERROR combined);*
*Award **[1]** for using RIGHT() up to the location of the space if there is no error;*
*Award **[1]** for correctly using FIND() twice to locate the second space when three names are present;*

Example answers
Use IFERROR to return one of two answers;
and handle any potential error;
Use RIGHT with FIND to return the surname if only two names;
Use RIGHT with two FIND functions to return the surname if there are two names;

```
= IFERROR(RIGHT(A2,LEN(A2)-FIND(A2," "),
          FIND( A2," ")+1)),
          RIGHT(A2,LEN(A2)-FIND(A2," ")))
```

Award full marks if any correct formula is given.

5. (a) *Award **[2]** max.*
*Award **[1]** for showing some correct calculations;*
*Award **[1]** for correct answer;*

Example answer:
KS = ((60 - 35) / (85 - 35)) * (91 - 40) + 40
KS = (25 / 50) * 51 + 40
KS = 0.5 * 51 + 40
KS = 25.5 + 40
KS = 65.5

(b) *Award **[3]** max.*
*Award **[1]** for using looping 5 times or specified number of students;*
*Award **[1]** for using the correct formula;*
*Award **[1]** for correctly rounding the result; // accept* `ROUND(MINISTRY[N])`
*Award **[1]** for correcting a score over 100 to 100;*

Example answer:

```
loop N from 0 to 4
MINISTRY[N] = ((SCORES[N]-35)/(85-35))*(91-40) + 40
    MINISTRY[N] = ROUND(MINISTRY[N],0)
    if MINISTRY[N] > 100 then
                                    MINISTRY[N] = 100
    end if
end loop
```

Note: Allow comparable versions of the formula like

```
MINISTRY[N] = (SCORES[N]-35)*(51/50) + 40
```

Note: The question is specific about the rounding, therefore other conversion functions like `INT(), CEIL(), TRUNC()` *are not correct.*

(c) *Award **[3]** max.*
*Award **[1]** for validation check;*
Award [2] for amplification statements;

Example answers:
Consistency checks / logical checks / relational checks;
Comparing student data across different data entries/subjects;
A student with a high overall grade and a low mark in a core subject might flag as an inconsistency;

Completeness checks;
Ensure that all required fields, such as student names and marks have been entered for every subject;
Missing score or student name in any subject would be flagged;

Length Consistency checks;
Check that the two lists are the same length;
Missing data would be flagged;

Duplication checks;
Ensure the same student isn't listed multiple times (with different marks);
This involves cross-referencing student IDs/names to detect duplicate entries;

Cross-validation with historical data/temporal consistency checks;
Student data compared with previous years;
A drastic change could trigger a review;

Statistical outlier detection;
Advanced statistical techniques could detect outliers in the data;
Machine learning algorithms/data mining might trigger a review;

Note: Do not award across clusters.
Note: Do not accept range check or type check – these are given in the stem
Note: Accept other correct answers.

(d) *Award **[4]** max.*
*Award **[1]** for initialising A_TOT to 0; // or any other identifier*
*Award **[1]** for using an if statement to check >=91; // allow >90*
*Award **[1]** for incrementing A_TOT;*
*Award **[1]** for correctly calculating the percentage;*

```
A_TOT = 0
loop N from 0 TO 4
    if MINISTRY[N] >= 91 then            // accept SCORES[N]
        A_TOT = A_TOT + 1
    end if
end loop
A_PERCENT = A_TOT/5*100
```

6. (a) *Award **[2]** max.*
*Award **[1]** for addressing real-time;*
*Award **[1]** for addressing simulation;*

Example answers:
A system that responds to data inputs immediately/minimal delay.
An emulation / representation of a real-world process or system for analysis or training.
An emulation of a real-world process that immediately responds to data inputs.

Do not accept 'simulation' to define real-time simulation

(b) *Award **[3]** max.*
*Award **[1]** for each variable.*

Example answers:
Academic scores;
School rating/resources;
School inclusion/selection policy;
Class sizes/student-teacher ratio;
Previous academic performance;
Language proficiency/percentage of EAL students;
Curriculum rigour;
Access to technology / internet;
School environment/policies on behaviour management;
Student health records;
Students' sleeping hours;
Socioeconomic status;
Teacher quality;
School location;
Parental engagement;
Extracurricular participation;

Do not accept:
School;
Day / date;

(c) *Award **[4]** max.*
*Award **[1]** for category/hardware type, and **[1]** for purpose.*

Example answers:
Data storage servers/cloud-based infrastructure/SANs; // accept memory on BOD
To process and store large volumes of data in real-time;

Networking equipment/switches/routers/digital lines;
To transmit data to the ministry;

End-user/school computers/tablets;
Hardware used to record attendance / take the tests;

Data security hardware/firewalls;
To protect data during transmission and storage;

Real-time analytics engines/specialist hardware;
For running real-time analytics on incoming data streams;

Power supply systems/UPS;
To maintain system operation during power failures.

Do not accept categories that do not relate to the size of this simulation, like:
CPU;
RAM;

*Mark as **[2]** + **[2]***

(d) *Award **[5]** max.*
*Award **[2]** for discussing one positive aspect;*
***Award [2]** for discussing one negative aspect;*
*Award **[1]** for a reasoned conclusion;*

Example answers:
Cluster A - Effective versus Ineffective test-case based on representative school
Sufficiently large sample population / Small school population;
Diverse sample population (culture, gender, etc.) / Homogenous population;
Varied attendance patterns / Atypical attendance patterns;
Range of academic performance / Atypical academic performance;
Varied socio-economic status / abnormally high or low socio-economic status;
Conclusion: Effective if representative because it provides generalisation / limiting generalisability if not representative;

Cluster B - Effective versus Ineffective based on data quality
Sufficiently large data population / Small data population;
Accurate attendance records / Inaccurate or incomplete attendance records;
Robust assessment procedures for attaining student scores / Inconsistent procedures for assessing student performance;
Moderated student scores for accuracy / Lack of accuracy in student scores;
Recorded scores may not translate to Ministry normalised scores effectively;
Conclusion: Effective if high data quality and sample size / Ineffective if poor quality data or limited data sample;

Cluster C - Effective versus Ineffective based on feedback mechanism:
School is committed to providing feedback data/School is not committed to supporting the ministry;
School is willing to change its approach to support model improvement / School is unwilling to change how it operates;
School inputs data promptly and timely / School fails to record attendance or student scores in a timely fashion;
Student scores have been moderated/standardised before submission / Student scores vary significantly between teachers;
Feedback from school is similar to previous years / Processes changed so data differs significantly from historical data;
Conclusion: Effective if feedback mechanisms improve the simulated model / Ineffective if feedback is not designed to support the model.

Note: Do not award across clusters.
Note: Accept a comparison of predicted data with real-life data for 2 marks max.

(e) *Award **[4]***
*Award **[1]** for stating a factor,*
***Award [1]** for an amplification.*

The number of data points;
More data points need more memory;

High resolution heatmaps;
Require more memory;

Colour depth (e.g. 8-bit, 16-bit, 24-bit) allows greater representation;
Higher colour depth requires more memory;

Matrix size (e.g. 1024 x 1024);
The higher the matrix the more memory;

The style of rendering impacts memory usage;
Some libraries use anti-aliasing or render optimisation;

The file type (raw bitmap/vector) affects memory;
Compression and format (e.g. jpg, png, svg).

*Mark as **[2]** + **[2].***

Option C — Web science

7. (a) *Award **[1+1, max 2]***

Resource location ;
host/host name,
domain,
sub domain,
Path,
parameters,
Query or query string,
Fragment.
protocol/scheme
top-level domain (TLD)
port number

(b) *Award **[1+1, max 2]***

Navigation / access
provide navigation between web pages/resources;
retrieve/fetch web pages/resources;
provide access to internet/web services;

Communication / protocols
send HTTP/HTTPS requests and process responses;
apply protocols such as HTTP and HTTPS to enable communication;
resolve domain names using DNS;
Rendering / content support
display/render web pages;
interpret/render HTML, CSS, XML;
support active content such as JavaScript;
support multimedia content/plugins;

Security
provide authentication and encryption features;
check/verify SSL/TLS certificates;
block unsafe/malicious websites, scripts or downloads;

Standards / compatibility
support web standards to ensure compatibility across websites and devices.

The web browser manages cookies;
for example, it stores, sends, blocks, deletes, or controls cookies used by websites to remember user preferences, login sessions, or tracking information.

(c) Award [3 max].

Award [1] for identifying the importance/purpose of protocols.
Award [1] for explaining how protocols support communication/interoperability/reliability/security.
Award [1] for further development, illustration or example.
Responses should include explanation/reasoning, not simple identification.

Rules / standards
a set of rules/procedures that devices and systems must follow;
define how data is formatted, transmitted and received
standardize communication between systems/devices;

Communication / interoperability
allow two or more devices/elements of a communication system to exchange/transport information;
enable interoperability between different hardware, software and networks;
ensure systems can communicate correctly even if developed by different manufacturers/vendors;
without protocols, systems/devices may not understand each other's messages or may interpret data incorrectly;

Reliability / data management
provide error detection and correction;
provide flow control;
help ensure accurate and reliable data transmission;
manage sequencing/order of transmitted data packets;

Security
support
secure communication/data transfer;
allow encrypted communication/authentication (eg HTTPS).
help ensure data integrity and security during communication.

(d) *Award **[1+1, max 2]***

Structure / organization
separates data and presentation/layout;
organizes data hierarchically/tree structure using parent and child elements;
uses nested elements/tags;

Extensibility / flexibility
extensible;
users can create/define their own tags;
allows custom data structures/descriptions;

Compatibility / portability
platform independent;
software/hardware independent;
supports interoperability/data exchange between different systems/applications;

Readability / usability
human readable;
machine readable;
plain text based

(e) Award [2 max].

Award [1] for describing the XML–XSL connection.
Award [1] for describing how the XSL processes/transforms/displays XML data.

Connection to data source (XML)
XML Sheet contains the data and the XSL sheet formats the presentation of the data; the browser processes the XML together with the XSL stylesheet to generate/display the webpage/HTML table;

The XML sheet connections to the XSL page;
`<?xml-stylesheet type="text/xsl" href="class_style.xsl"?>`

XSL sheets sets rules for template matching;
XSL stylesheet acts as a template for transforming the XML data into an HTML page;`<xsl:template match="/">`

XSL sheet Selects elements from XML file;

```
<xsl:for-each select="class/student">
<tr>
  <td><xsl:value-of select="firstname"/></td>
  <td><xsl:value-of select="surname"/></td>
  <td><xsl:value-of select="classname"/></td>
</tr>
</xsl:for-each>
```

The XSL sheet loops through the data in the selected elements;
`<xsl:for-each select="class/student">`
mark the start of the loop
`</xsl:for-each`
Marks the end of the loop

(f) *Award **[1+1 +1, max 3]***

Location
implemented in the transport layer;

Reliability
reliable protocol;
uses acknowledgements to confirm receipt of packets;
retransmits lost/missing packets;
guarantees delivery of data;
performs error checking/error recovery;
uses checksums/error checking to detect corrupted data during transmission;

Connection-oriented communication
connection-oriented;
establishes a connection between sender and receiver before transmission;
maintains the connection during communication;
uses a handshake/three-way handshake to establish the connection;
supports full duplex communication;

Sequence / packet management
uses sequence numbers in packets/segments;
ensures packets are reassembled in the correct order;
resends missing/out-of-order packets.

Flow control
uses flow control to prevent the receiver being overwhelmed

Congestion control
uses congestion control to manage/reduce network congestion;

Data segmentation
breaks data/documents into smaller packets/segments;
reassembles packets/segments at the destination using sequence information

Process / application communication
uses port numbers to support process-to-process/application-to-application communication.

8. (a) *Award **[1]** for deep web and **[1]** for surface web comparison/difference.*

Accessibility;
surface web content is publicly accessible/searchable using standard search engines;
deep web content is not easily accessible/searchable through standard search engines;

Indexing;
surface web pages are indexed by search engines;
deep web content is typically not indexed/searchable by standard search engines;
deep web content may require specialized/internal search tools or direct access;

Composition/content;
surface web contains publicly available/static webpages such as blogs, news and public websites;
Deep web content may be behind authentication, forms, paywalls or databases, while surface web pages are publicly discoverable;

Access methods;
surface web is typically accessed using standard browsers/search engines;
deep web content may require authentication/login credentials and may sometimes require specialized tools/applications;

(b) *Award **[1]***

a computer program/bot/spider that automatically browses/crawls webpages to collect and index information for search engines.

(c) *Award **[1+1+1, max 3]***

Starting point / crawling
the crawler starts at a seed/designated starting page/URL;
the crawler downloads/accesses the webpage content;

Indexing / analysis
the crawler reviews/analyzes page content;
the crawler categorizes/indexes webpage information;
the crawler identifies keywords, metadata, links and other content;
indexed content is stored for retrieval/ranking by the search engine;

Link traversal
the crawler follows hyperlinks/URLs to other webpages;
the crawler recursively repeats the process on linked pages;
may use breadth-first or depth-first crawling approaches;
Restrictions / control
the crawler may follow rules defined in robots.txt;
robots.txt may restrict/prevent crawling/indexing of certain pages/directories.

(d) Award [2+2] per cluster/advantage: [***max 4]***]

[1] for each identification of a valid advantage;
[1] for each description/development.

Distributed load;
Distributed processing / load distribution
workload is distributed among multiple crawlers/servers;
reduces the processing burden on a single crawler/system;
improves efficiency/performance;

Reduced latency / speed
reduced latency when crawlers are geographically distributed;
pages can be crawled closer to their source/server location;
simultaneous crawling increases crawling speed/throughput;

Scalability
can crawl/index very large portions of the web;
easier to scale by adding additional crawlers/resources;
improves time efficiency when handling large datasets/websites;

Resource management / load balancing
supports redistribution/balancing of crawling tasks/resources;
improves coverage and resource utilization;
prevents some crawlers/servers from becoming overloaded;

Fault tolerance / reliability
if one crawler fails, other crawlers can continue operating;
redundancy improves reliability/availability of the crawling system;

Cost efficiency / operational efficiencyparallel crawling can reduce
operational/time/infrastructure costs by processing multiple pages simultaneously;
improved throughput may reduce the resources and time required for large-scale crawling;

(e) Award *Max **[5].***

Content quality / relevance
developing high-quality, relevant and valuable content/pages;
reducing duplicate/repeated/thin content;
maintaining accurate and up-to-date content;

Keywords / metadata
using appropriate keywords within page content/body text;
using appropriate keywords in meta tags/descriptions;
using meaningful meta descriptions;
avoiding excessive keyword stuffing;
using keyword-rich but relevant metadata;

Page structure / semantic HTML
developing suitable page structure/semantic HTML;
using heading tags appropriately (H1, H2, H3);
using alt tags for images/accessibility;

Links / authority
using high-quality incoming and outgoing links/backlinks;
links from authoritative sites may improve PageRank/HITS authority;
internal linking may improve navigation and indexing;

User experience / performance
optimizing user experience/navigation;
fast page load times;
optimized page settings/configuration;
reduced/compressed image sizes;
progressive/lazy image loading;

Responsive/mobile optimization
responsive page design;
mobile optimization/mobile-friendly pages;

Maintenance / indexing
regular updates of page content;
frequent/efficient search engine re-indexing/crawling;

Marks	Descriptor
0	The response does not reach a standard described by the descriptors below.
1	Identifies one basic white hat SEO technique with little or no development. Response may be vague, generic or partially accurate.
2	Identifies one valid white hat SEO technique and provides limited description or simple benefit. Limited understanding of SEO impact is demonstrated.
3	Describes two valid white hat SEO techniques OR develops one technique well. Some understanding of how the techniques improve search engine ranking/visibility is shown. Limited evaluation may be present.
4	Describes and evaluates two valid white hat SEO techniques. Response includes clear development, impact or effectiveness of the techniques. Terminology is generally accurate and examples/details may be included.
5	Provides a well-developed evaluation of two valid white hat SEO techniques. Demonstrates clear understanding of effectiveness, limitations, trade-offs or impact on ranking/user experience. Uses accurate technical terminology and relevant supporting detail/examples throughout.

9. (a) *Award **[1]** for each reason stated and **[1]** for each amplification. Max **[4].***

Device focus / integration
mobile computing focuses on the use of portable/mobile devices such as phones, tablets and laptops;
ubiquitous computing is integrated into the environment/everyday objects (eg IoT devices, smart homes, wearables, AR systems);

Visibility / interaction
mobile computing is typically user-visible and device-centered;
ubiquitous computing is often embedded/integrated/invisible to the user;
mobile computing typically involves direct user interaction through a device/screen;
ubiquitous computing is often proactive/context-aware, with the environment responding automatically to the user;

Connectivity / scope
mobile computing focuses primarily on an individual portable device operating anywhere/anytime;
ubiquitous computing emphasizes interconnected systems/devices embedded throughout the environment;

Context awareness / adaptation
ubiquitous computing is often context-aware and can adapt to the user/environment;
mobile computing does not necessarily require environmental/context awareness;

Mobility
mobile computing emphasizes portability and mobility;
ubiquitous computing emphasizes seamless integration into daily life/environment rather than portability alone.

(b) *Award **[1]** for reason stated and **[1]** for amplification. Max **[2]***

Openness / transparency
open standards are publicly available and transparent;
access to the specifications is not restricted by a developer/company;
developers can implement the standard without needing proprietary permission;

Common standards / formats
open standards use common specifications, protocols, APIs and/or data formats;
common formats/APIs make data exchange possible without custom conversion;
systems can exchange files/data using shared formats;

Cross-platform compatibility / interoperability
different platforms, devices, software and services can communicate efficiently;
cross-platform compatibility is supported;
systems developed by different developers/companies can work together;

Ease of implementation / integration
open standards are easier to implement because specifications are accessible;
they support easier integration between systems/applications;
community support can assist implementation and compatibility;

Sustainability / development community
open standards may be maintained by a community;
a range of developers can develop compatible solutions;
community maintenance can support long-term interoperability;

Financial / cost
open standards can reduce costs because organizations may avoid proprietary formats, licences or custom conversion systems.

(c) Award *Max **[6].***

Reduced centralized control
no single entity/corporation has full control over user data or content;
reduced dependence on a central provider/platform;
reduced risk of censorship/control by a single organization;

Increased user autonomy:
User autonomy / privacy
users can choose which server/service/community to join;

users may have greater control over personal data;users may have greater control over privacy settings and online identity;

Portability / interoperability
open standards may allow users to move more easily between servers/services;

data portability increases user choice;
users are not locked to a single provider/platform;
services using common/open standards may support interoperability between networks;

Governance / communities
local governance of services/servers may be possible;
communities can form around shared interests, beliefs or principles;
different communities may create their own moderation/governance rules;

Security / resilience
distributed/decentralized systems may be more resilient because data/services are not centralized;
failure of a single server/node may not bring down the entire network;
reduced single point of failure;

Marks	Descriptor
0	The response does not reach a standard described by the descriptors below.
1–2	Identifies one or two basic characteristics/advantages of decentralized or distributed social networks. Response is limited, vague or partially accurate with little development.
3–4	Describes some relevant characteristics/advantages of decentralized or distributed social networks. Response demonstrates some understanding of concepts such as autonomy, interoperability, governance or resilience. Development is present but may be uneven or lacking detail.
5–6	Provides clear and well-developed descriptions of multiple relevant characteristics/advantages of decentralized or distributed social networks. Demonstrates good understanding of concepts such as reduced centralized control, portability/interoperability, governance, privacy/autonomy and resilience. Uses accurate terminology and supporting detail/examples where appropriate.

(d) Award *Max **[4].***

Lack central authority to enforce copyright and IP rules uniformly;

The different sites may have different policies and levels of checking;
Application of the rules and laws is inconsistent

Requirements to attribute work may vary from site to site;

Attribution and ownership tracking the original source of digital content becomes difficult leading to risk of unauthorised reuse or re-hosting without credit.

Requirements to seek permission to use works may vary;

Monitoring and moderation;
Site may be small, insufficient people to efficiently check and deal with copyright.
Volunteers for monitoring rather than paid employees or automated processes

Legal requirements of local/national laws;
Compliance with USA the Digital Millennium Copyright Act (DMCA) or the EU's Copyright directive.
Server/service may be in a different jurisdiction and there fore have different laws applied.

Tools and processes;
may be inconsistent
May lack common tools available to check copyright.

Marks	Descriptor
0	The response does not reach a standard described by the descriptors below.
1	Identifies a basic challenge related to intellectual property in distributed social networks. Response is limited, vague or partially accurate.
2	Describes one relevant intellectual property challenge OR identifies multiple basic challenges. Some understanding of distributed social networks is demonstrated.
3	Discusses relevant intellectual property challenges faced by distributed social networks. Response includes development/explanation of issues such as copyright enforcement, ownership of content, moderation difficulties or cross-jurisdictional/legal challenges.
4	Provides a clear and balanced discussion of multiple intellectual property challenges faced by distributed social networks. Demonstrates good understanding of issues such as copyright infringement, ownership/control of content, decentralized moderation, enforcement difficulties and legal/jurisdictional complexity. Uses accurate terminology and relevant supporting detail/examples where appropriate.

Option D — Object-oriented programming

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

A class takes less (no) memory (at runtime) / class is allocated memory for static variables;
Objects are allocated memory / each object has memory to store instance variables;
Objects are stored in heap whereas class is stored elsewhere;

(b) *Award **[1 max].***
*Award **[1]** for an accessor with the correct prefix 'get' followed by correct instance variable;*
Example – `getType(), getLevel(), getSpeed(), getHyperdrive()`

Note: Accept the accessor with return type.

(c) *Award **[2 max].***
*Award **[1]** for identifying the purpose, **[1]** for expanding it;*

A (public) mutator method allows access to a private (instance) variable;
for the purpose of changing its value from outside the class;

mutator methods provide controlled access;
to change the attributes of an object;

mutators are used to update a private variable's value;
from outside the class scope;

mutators facilitate encapsulation;
as private data members cannot be modified directly;

Note: Do not allow mixing the points from different clusters.

(d) *Award **[3 max].***
*Award **[1]** for* `public void`.
*Award **[1]** for correct parameter.*
*Award **[1]** for correct assignment;* // keyword `'this'` is not required.

```
public void setType(String type)
{
    this.type = type;
}
```

(e) *Award **[4 max].***
*Award **[1]** for* `extends Ship;`
*Award **[1]** for the additional attribute* `stealth` *of* `boolean` *type;*
*Award **[1]** for correct constructor (parameters + use of* `super()`*);*
*Award **[1]** for setting* `stealth` *to* `false;` // keyword 'this' is not required.
*Award **[1]** for any inclusion of accessor/mutator methods for* `stealth;`

Example 1:

```
public class PowerShip extends Ship
{
    private boolean stealth;   // accept stealth = false;

    public PowerShip(String t, int l, int s, boolean h, int si)
        super(t, l, s, h, si);
        this.stealth = false;
    }

    public boolean getStealth() { return stealth; } // accept without body
    public void setStealth(boolean st) { stealth = st; } // accept without body
}
```

(f) *Award **[2 max]**.*

change the datatype;
to int/String/char datatype; // accept other datatypes

change the name of hyperdrive;
to another name such as enginetype;

Note: Do not accept addition of a new variable. Do not allow mixing the points from different clusters.

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

Abstraction;
Encapsulation;
Polymorphism;
Inheritance;

Note: Accept modularity, classes and objects, aggregation, composition, and association.

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

Encapsulation / Data hiding;

(c) *Award **[2 max]**.*

Encapsulation

Encapsulation protects data / enhances security / enables data hiding;
because attributes are declared private / data can only be accessed or changed through controlled methods;

Inheritance

Inheritance promotes code reuse / reduces development time;
because a subclass can extend a superclass and reuse existing attributes and methods;

Inheritance reduces maintenance overhead;
because changes to common code in the superclass can be inherited by subclasses;

Polymorphism

Polymorphism allows the same method call to produce different behaviours;
because subclasses can override methods from the superclass;

Abstraction

Abstraction hides implementation details / reduces complexity;
because methods can be used without knowing how they are implemented;

Note: Accept advantages of other valid OOP features such as modularity, classes and objects, aggregation, composition, and association.

(d) *Award **[2 max].***

Encapsulation
Encapsulation can make code longer / more complex;
because accessor and mutator methods may be needed to access or change private attributes;

Encapsulation can make debugging harder;
because the data is hidden inside the class and cannot be accessed directly from outside the class;

Inheritance
Inheritance can increase dependency between classes;
because changes in the superclass may affect all subclasses that inherit from it;

Inheritance can make the program harder to understand;
because behaviour may be spread across both superclass and subclass code;

Inheritance can lead to inappropriate code reuse;
because a subclass may inherit attributes or methods that it does not need;

Polymorphism
Polymorphism can make code harder to trace / debug;
because the method that runs depends on the object type at runtime;

Polymorphism can make programs harder to understand;
because the same method call may produce different outcomes for different objects;

Abstraction
Abstraction can hide too much implementation detail;
because programmers using the class may not understand how the method works internally;

Abstraction can make errors harder to locate;
because the user of the method may only see the interface and not the underlying implementation;

Note: Accept disadvantages of other valid OOP features such as modularity, classes and objects, aggregation, composition, and association.

12. (a) *Award **[6 max].***
*Award **[1]** for correct return type;*
*Award **[1]** for declaring and initialising* `sum` *to 0;*
*Award **[1]** for correct loop through* `refuel`*; // accept '*`for`*', '*`while`*' or enhanced '*`for`*';*
*Award **[1]** for checking for* `null` *in* `refuel`*;*
*Award **[1]** for attempt to calculate* `sum`*;*
*Award **[1]** for calculating* `sum` *using correct accessor (*`getSize() or getType()`*);*
*Award **[1]** for returning the* `sum`*;*

Example 1:

```
public int totalFuel()           // for loop
{
    int sum = 0;
    for (int i=0; i<100; i++)
    {
        if (refuel[i] != null)
        {
            sum = sum + refuel[i].getSize();
        }
    }
    return sum;
}
```

Example 2:

```
public int totalFuel()          // while loop with 2 conditions
{
    int sum = 0;
    int i = 0;
    while((i < 100) && (refuel[i] != null))
    {
        sum = sum + refuel[i].getSize();
        i++;
    }

    return sum;
}
```

Example 3:

```
public int totalFuel()          // enhanced for loop
{
    int sum = 0;
    for (Ship object: refuel)
    {
        if (object != null)
        {
            sum = sum + object.getSize();
        }
    }
    return sum;
}
```

(b) *Award **[5 max]**.*
*Award **[1]** for using* `totalFuel();`
*Award **[1]** for a correct test to check if there is enough fuel left;*
*Award **[1]** for declaring and initialising a counter;*
*Award **[1]** for correct loop to find the first empty location in the array;*
*Award **[1]** for assigning the* `aShip` *in the location found; // even if loop incorrect.*
*Award **[1]** for outputting (not returning) an error message; // accept* **`IBIO.output`**.

Example 1:

```
public void addRefuelShip(Ship aShip)
{
    if (aShip.getSize() <= 80-totalFuel())
    {
        int last = 0;
        while((last < 100) && (refuel[last] != null))
        {
            last++;
        }
        refuel[last] = aShip;
    }
    else
    {
        System.out.println("Maximum refuel limit reached.");
    }
}
```

Example 2:

```
public void addRefuelShip(Ship aShip)
  {
    if(totalFuel() + aShip.getSize() <= 80)
    {
      boolean added = false;
      for (int i = 0; i < refuel.length && !added; i++)
      {
        if (refuel[i] == null)
        {
          refuel[i] = aShip;
          added = true;
        }
      }
    }
    else
    {
      System.out.println("Maximum refuel limit reached.");
    }
  }
```

(c) *Award **[4 max].***
*Mark as **[2]** + **[2]**.*

Provides robust / reliable code
Libraries contain code that has usually been thoroughly tested;
therefore this can reduce errors in the program;

Speeds up development
Programmers do not need to reinvent existing algorithms or classes;
therefore the program can be developed more quickly;

Reduces maintenance overhead
Library objects may be updated and maintained by the library developers;
therefore programmers may spend less time maintaining their own custom code;

Provides reusable objects
Libraries provide ready-made objects/classes that can be used in different programs;
therefore programmers do not need to write the same functionality repeatedly;

Note: Do not allow mixing the points from different clusters. Do not allow easier/faster without any further description in context of libraries.

13. (a) *Award **[4 max].***
*Mark as **[2]** + **[2]**.*

It supports characters from all languages / uses Unicode;
enables players' names/chat messages in multiple languages;

Locale-sensitive APIs;
adjusts game behaviour based on the player's locale;

supports region-specific date/time/currency formats;
allowing players from different countries to use familiar formats;

Supports String translation / translation of text;
allows game text/messages to be translated without changing code;

Bidirectional text support;
supports left-to-right and right-to-left text;

sorting based on language-specific rules;
allows correct ordering of player names/messages;

Externalized messages and string resources / resource bundles;
stores text outside the code for translation purposes;

Multilingual I/O handling;
handles input/output in multiple languages;

Note: Accept other valid answers as long as they enable internationalization.

(b) *Award **[5 max].***
*Mark as **[2]** + **[2]** + **[1]**.*
*Award **[1]** for identifying an obligation.*
*Award **[1]** for an amplification / reason*
*Award **[1]** for identifying a second obligation.*
*Award **[1]** for an amplification / reason.*
*Award **[1]** for providing a balanced conclusion.*

The obligation to respect privacy;
Certain info (like address or phone number) should not be allowed to be posted in a chat;

The obligation to provide security;
The chat should be monitored (as to avoid bullying) by features to monitor the chat automatically;

The obligation to promote accessibility;
Programmers could include features such as text-to-speech, etc.;

The obligation to promote inclusivity;
allowing players from different cultures/backgrounds/identities to participate comfortably;

The obligation to protect children from addiction;
Programmers could incorporate down time / limit the duration of the gaming experience;

Conclusion:
Programmers should be aware of these responsibilities / obligations in order to protect adolescents;

Note: Please accept other valid obligations with some amplification/reason.