

Markscheme

November 2018

Computer science

Higher level

Paper 2

34 pages

This markscheme is the property of the International Baccalaureate and must **not** be reproduced or distributed to any other person without the authorization of the IB Global Centre, Cardiff.

Option A — Databases

1. (a) *Award up to [3 max].*

Award [1] for a definition of an information system, [1] for a definition of a database and [1] for the relationship between the two.

An information system is the collection of software, hardware, networking infrastructure, human resources and databases that provide the storage, processing and communication/distribution of information;

A database is an organized collection of data (organized so that its contents can easily be accessed, processed, and updated);

A database is a subset of the information system;

[3]

(b) *Award up to [8 max].*

Award up to [2 max] for each of the four properties,

Award [1] for an explanation of the property and [1] in relation to the money transfer transaction.

Example answers:

Atomicity:

Ensures that either all changes to data are performed as if they are a single operation or none of them are performed;

In moving money from one account to another, it ensures that if a debit is made successfully from one account then the corresponding credit is made to the other account;

Consistency:

Ensures that data is in a consistent state when a transaction begins and when it ends / ensures that only valid data is written in the database;

In transferring money from one account to another, it ensures that the total value of money in both the accounts is the same at the beginning and at the end of each transaction;

Isolation:

Ensures that intermediate state of a transaction is invisible to other transactions;

In moving money from one account to another the isolation property ensures that another transaction might see one account or the other but cannot see both accounts;

Durability:

Ensures that after a transaction successfully completes, changes to data persist and are not undone even in the event of a system failure;

In moving money from one account to another durability property ensures that the changes made to each account are permanent;

[8]

- (c) *Award up to [6 max].*
Award [1] for identifying an advantage of these databases being open to interrogation and an additional [1] for each development of that advantage up to [3 max].
Award [1] for identifying a disadvantage of these databases being open to interrogation and an additional [1] for each development of that disadvantage up to [3 max].
Award [1] for a valid conclusion.
If there is no conclusion, award [5 max].

Advantages:

Sharing the information may enable it to be made accessible to organizations such as the Police or the Government that may not have access to it;
This means that it could be used to detect fraudulent financial activities;
Leading to reduced insurance or health insurance premiums as monies are not lost from the system that create artificially high premiums;

Disadvantages:

Misuse of information / security issues;
Information from this database may enable the account details of the clients to be shared with third parties;
Who may use this information to take money from their accounts;

Conclusion:

Although the information may be shared, the overall benefits from a reduction in fraudulent crime or inflated insurance premiums is greater than the potential for this information to get into the wrong hands;

[6]

2. (a) *Award up to [1 max].*
A database management system (DBMS) is a set of programs that allows to read, store, change /extract data in a database; [1]
- (b) *Award up to [3 max].*
Award [1] for each characteristic of logical schema identified up to [3 max].
All attributes for each entity in database are specified;
The primary key for each entity is identified;
Includes all entities and relationships among them;
Keys identifying the relationship between different entities (foreign keys) are specified;
Normalization occurs at logical level; [3]
- (c) *Award up to [2 max].*
Award [1] for each identifying the role of Data Definition Language (DDL) and [1] for an expansion or explains how it may be used up to [2 max].
The Data Definition Language (DDL) are commands which define the different structures/objects in a database;
Database administrators use these commands during the setup and/or removal of a database and database objects (such as tables, indexes, and users); [2]
- (d) *Award up to [2 max].*
Award [1] for each task that a database administrator carries out to ensure the security of the database up to [2 max].
DBA sets access levels and passwords;
DBA manages back-up procedures (allow examples of this);
DBA establishes a recovery plan for the database in disaster case; [2]

3. (a) (i) *Award up to [1 max].*
 Real number/float;
 Accept currency. [1]
- (ii) *Award up to [3 max].*
 Award [1] for selecting correct field from the relation.
 Award [1] for correct supplier name comparison.
 Award [1] for correct checking the price range.
 Award [1] for correct logical operations. [3]

Example answer 1:

```
SELECT Prod_Name FROM Greengrocer
WHERE Supp_Name == 'Veggy Co.' AND Prod_Price >= 4.00 AND
Prod_Price <= 10.00
```

Example answer 2:

```
SELECT Prod_Name FROM Greengrocer
WHERE NOT (Prod_Price <4.00 OR Prod_Price >10) AND
Supp_Name == 'Veggy Co.'
```

Example answer 3:

```
SELECT Prod_Name FROM Greengrocer
WHERE NOT (Prod_Price <4 OR Prod_Price >10 OR Supp_Name !=
'Veggy Co.' )
```

Accept logically equivalent answers.

- (b) *Award up to [2 max].*
 Redundant data unnecessarily repeats in a relation;
 Supplier data (name, phone, contact) are repeated;
 For every item that is supplied by the same supplier;

For example,

Items P118, P122 and P220 have the same supplier (Veggy Co)

Items P219, P111, P121 and P211 have the same supplier (Fruit and Veggie)

Note: Accept any correct example from the table given, such is the following:

<u>Prod_ID</u>	Prod_Name	Prod_Price	Supp_Name	Supp_Contact	Supp_Phone
P211	Apples	5.00	Fruit and Veggie	Ines Bond	542 9800
P121	Onions	3.00	Fruit and Veggie	Ines Bond	542 9800

[2]

- (c) (i) **Award up to [2 max].**
Award [1] for explaining a problem that may arise (such as inconsistency, inaccuracy / unnecessary additional/repeated work needed when updating).
Award [1] for including example with reference to the relation given.

Example answer:

When inserting a tuple supplied by the existing supplier, for example, inserting:

P189	Potatoes	9.90	Veggy Co.	234 7654	Mia Abiss
------	----------	------	-----------	----------	-----------

All data about supplier should be accurately entered;
 And consistent with the data about this supplier in all other tuples;

[2]

- (ii) **Award up to [2 max].**
Award [1] for explaining a problem that may arise (such as inconsistency, inaccuracy).
Award [1] for including example with reference to the relation given.

Example answer:

Deleting a tuple could remove data which is not intended to be lost;
 For example; deleting the following tuple.

P265	Tomatoes	7.45	New Fruits	John Smith	677 7895
------	----------	------	------------	------------	----------

Which is the only tuple in the relation which holds data about New Fruits suppliers (all data about this supplier will be lost);

[2]

- (iii) **Award up to [2 max].**
Award [1] for explaining a problem that may arise (such as inconsistency, inaccuracy/unnecessary additional/repeated work needed when updating).
Award [1] for including example with reference to the relation given.

Example answer:

Modifying/changing the phone number for Fruit and Veggie. would require that each and every tuple containing item supplied by Fruit and Veggie should be modified (unnecessary work);

<u>Prod_ID</u>	Prod_Name	Prod_Price	Supp_Name	Supp_Contact	Supp_Phone
P211	Apples	5.00	Fruit and Veggie	Ines Bond	542 9800
P121	Onions	3.00	Fruit and Veggie	Ines Bond	542 9800

Otherwise the data in the relation would be inaccurate (for example, would contain different/incorrect phone numbers for the same supplier);

<u>Prod_ID</u>	Prod_Name	Prod_Price	Supp_Name	Supp_Contact	Supp_Phone
P211	Apples	5.00	Fruit and Veggie	Ines Bond	542 9800
P121	Onions	3.00	Fruit and Veggie	Ines Bond	542 1811

[2]

- (d) *Award up to [8 max].*
Award [1] for explaining/clearly showing each of the steps.
Award [1] for identifying two entities (Product and Supplier) in Greengrocer.
Award [1] for choosing/identifying a primary key in relation Supplier.
Award [1] for clearly showing that the relation must be in 1NF and then could be put to 2NF.
Award [1] for showing that each tuple in 2NF has a primary key.
Award [1] for showing that in 2NF all attributes are on the whole dependent of the primary key.
Award [1] for introducing new relations or using a foreign key.
Award [1] for showing that the relation must be in 2NF and then could be put to 3NF.
Award [1] stating that there are no transitive dependencies in 3NF.

Note: *Because there is significant inconsistency in textbooks' accounts of precisely what constitutes 1NF and 2NF, award marks depending on the explanation. Please see the two different example answers.*

Example answer 1:

Given relation

Greengrocer(Prod_ID, Prod_Name, Prod_Price, Supp_Name, Supp_Phone, Supp_Contact)

There are two entities in this relation (Product and Supplier).

Product(Prod_ID, Prod_Name, Prod_Price)

Supplier(Supp_ID, Supp_Name, Supp_Phone, Supp_Contact)

Primary key (Supp_ID) in relation Supplier is added to make sure that it is unique because it might be that two suppliers have the same name.

Product(Prod_ID, Prod_Name, Prod_Price, Supp_ID)

Supplier(Supp_ID, Supp_Name, Supp_Phone, Supp_Contact)

To construct to 2NF relation must be in 1NF;

Each tuple in 2NF must have a primary key;

And all attributes in the 2NF must be dependent on the whole of the primary key.

New relation is introduced.

Product(Prod_ID, Prod_Name, Prod_Price)

Supplier(Supp_ID, Supp_Name, Supp_Phone, Supp_Contact)

ProdSupp(Prod_ID, Supp_ID)

There are two different contacts with the same name (Mia Abiss) at the two different companies.

Assuming that each contact has a unique phone number, a new relation could be introduced in which Supp_Phone is a key.

Supplier(Supp_ID, Supp_Name)
SupplierContact(Supp_Phone, Supp_Contact)
Product(Prod_ID, Prod_Name, Prod_Price)
ProdSupp (Prod_ID, Supp_ID)

The relation is in the 2NF **and** there are no attributes which are not dependent on the key.

There are **no** transitive dependences so it is in the 3NF.

Example answer 2:

The table is already in 1NF because all values are atomic. There are no "fake" attributes/columns, like Supp_Phone1, Supp_Phone2;

So there are no repeating groups, either obvious or created by repeating columns for what is really the same attribute;

To be in 2NF, in addition to being in 1NF;

Every non-prime attribute must be dependent on the whole of every key;

To be in 3NF, in addition to being in 2NF;

All the attributes in a relation must be determined only by the key(s);

And there should not be any transitive dependences;

There are two entities (Supplier and Product);

Supp_ID should be introduced;

Supplier(Supp_ID, Supp_Name, Supp_Phone, Supp_Contact)
Product(Prod_ID, Prod_Name, Prod_Price, Supp_ID*)

Where underlining marks primary key(s) and *foreign key(s);

There are two different contacts with the same name(Mia Abiss)at the two different companies.

A new relation introduced in which Supp_Phone is a key;

SupplierContact(Supp_Phone, Supp_Contact, Supp_ID*)
Supplier(Supp_ID, Supp_Name)
Product(Prod_ID, Prod_Name, Prod_Price, Supp_ID*)

[8]

4. (a) *Award up to [4 max].*

Award [1] for each source of information up to [4 max].

For example,

Sources of travel information such as airports, ports, bus/train;

Sources of personal data about tourists (such as age, gender, *etc*);

Sources of social data about tourists (such as family status, occupation, economic circumstances, *etc*);

Sources of information about the tourism products and services these tourists booked such as:

- accommodation (hotels, camps, apartments);
- attractions (natural parks, palaces, museums);
- activities (yachting, biking, diving, fishing, surfing);

[4]

(b) *Award up to [4 max].*

Award [1] for each identifying a difference between a data warehouse and a database and [1] for an expansion up to [2 max].

Mark as [2] and [2].

Database is based on operational processing;

Whilst data warehouse is based on informational processing;

The operations on databases consist of transactions;

Whilst operations on data warehouse consist of queries;

Database is used for everyday transactions;

Whilst data warehouse is used for decision support;

Database mainly stores current data;

Whilst data warehouse stores historical data;

Data in the database is always up to date/accurate;

Whilst the data in data warehouse is maintained over time;

Data in the database is simple/primitive/detailed;

Whilst data warehouse holds summarized/consolidated data;

The type of access to data in database is read/write;

Whilst mostly read only access for the data stored in data warehouse;

Database size is smaller (for example in up to 10 GB);

Than the size of data warehouse (for example 100 GB to TB);

View of data in database is flat/relational;

Whilst the view of data in data warehouse is multidimensional;

Users (end-users) of the database are common users (for example customers, tourists, clerks, *etc*);

Whilst end-users of data warehouse are knowledge users (for example, analysts, managers, executives);

The number of database users is greater;

Whilst the data in data warehouse is used by smaller number of people;

[4]

- (c) *Award up to [3 max].*
Extract data

Award [1 max] for an explanation of how the extraction process can be used to address the problems related to data migration.

Pulling data from different source databases/ from various source systems (eg MS Access, SQL Server, Oracle, etc);

(Cleanse and) transform data (so that data is clean and useful for a purpose)

Award [1 max] for an explanation of how the transformation process can be used to address the problems related to data migration.

Trimming for white space/ proper data type/do some validation, etc;

Load data

Award [1 max] for an explanation of how the load process can be used to address the problems related to data migration.

Transfer data into data warehouse/ data mart/operational data store so it can be used for the mining processes within the data warehouse;

[3]

- (d) *Award up to [2 max].*

Award [1] for each identifying an ethical problem that may result from data mining and [1] for an expansion up to [2 max].

Data mining raises privacy concerns when people are traced;
Their actions are analyzed without their knowledge;

Data mining creates people/tourist/customer files with a tendency of judging and treating people on the basis of group characteristics;
Instead of on their own individual characteristics;

Data mining can provide information that could be used unethically;
For example, increasing profits by selling this information to others / or by targeting return tourists with information obtained on their previous visits;

[2]

- (e) *Award up to [4 max].
Award [2 max] for evidence that candidates understand how cluster analysis is used and [2 max] for an explanation related to tourism products/services.*

Cluster analysis:

Subdivide tourists into distinct subsets;
Where any subset may be selected as a market target;

Explanation related to tourism products/services:

Subsets could be “young tourists with limited budget”;
For this subset music festivals, sport could be advertised to ensure the hotel can maximize its revenue;

Subset of tourists traveling with young children;
Children activities / sports / games available / “kids hotels” could be advertised to ensure the hotel can maximize its revenue;

Subset of older tourists influenced by cost–value considerations and/or secure/or quiet environment;
So these hotels can be configured to specifically meet their needs and promoted only to these tourists such as advertise “stay 7 days and pay only 6 days” / advertise hotel located in the quiet place/secure place / at some distance from the center;

[4]

- (f) *Award up to [3 max].
Award [1] for each identifying the importance of link analysis in exploring patterns in data mining and [1] for each subsequent expansion up to [3 max].*

Link analysis technique is used to analyse connections/links between small instances of relational databases / only on required datasets of database/;
(But) it can also be applied on large number of databases to show the relationships between these databases;
Link analysis is important because it discovers new relations in relational databases / new patterns of interest / checks the similarity between the datasets / finds anomalies where old/known patterns are violated;

[3]

Option B — Modelling and simulation

- 5. (a) *Award up to [3 max].*
*Award [1] for **correctly** indicating type, and values expected, for each variable up to [3 max];*

```

“MeterType”, String/char; “old”, “new”;
“WaterVol”, Integer; 0–999999;
“InfraCharges”, Currency/Float /Double; 184.32–204.80;

```

[3]

- (b) Making use of both the previous and new readings;
 Calculating the difference between them; **[2]**

- (c) *Award up to [7 max] as follows:*
 Involving previous readings;
 Calculation of water supplied **[2]** – allow **[1]** if correct apart from the reset;
 Separate code for use of new meter;
 Correct reduction of infrastructure charge **[2]** – allow **[1]** for an attempt to calculate this;
 Calculation of final bill;

Example:

```

householdBill(OLD_WATER_IN, OLD_WATER_OUT) // the previous
                                                // meter readings

INFRACHARGE = 204.80
  if WATER_IN > OLD_WATER_IN // allows for possible
                              // resetting of meter
                              //reading to 0
      WATER_SUPPLIED = WATER_IN - OLD_WATER_IN
  else
      WATER_SUPPLIED = 1000000 - OLD_WATER_IN + WATER_IN
  end if

if METER_TYPE = "new" // possible reduction in sewage charges
                      // if new meter is used
  if WATER_OUT > OLD_WATER_OUT
      WATER_RETURNED = WATER_OUT - OLD_WATER_OUT
  else
      WATER_RETURNED = 1000000 - OLD_WATER_OUT + WATER_OUT
  end if
  if WATER_RETURNED > 0.95 * WATER_SUPPLIED
      INFRACHARGE = INFRACHARGE * 0.9
  end if
end if

BILL = WATER_SUPPLIED * 1.9 + INFRACHARGE

```

[7]

- (d) Changing to a new meter (if they have an old one);
 Not using the supplied water for watering the garden *etc*;
 Diverting rain water into the waste/sewage system; **[3]**

6. (a) The model is a static description of some rules;
In this case a list of appliances and their water consumption (min or max);
Whereas the simulation is the use of that model;
With input from real parameters (such as those that families may input); **[4]**
- (b) *Award up to [4 max].*
There may be missing data:
For example, watering plants (allow any suitable example);
The consequences, *eg* the totals will be underestimated;
The data may be inaccurate;
For example, the shower times may exceed the stated;
The consequence, *eg* the results of the simulation may not be valid; **[4]**
- (c) The answer shows the use of columns and rows;
The data from part (b) is included together with the addition of a `TimesPerWeek` variable;
Weekly total formula for each variable is included;
Overall weekly total has correct formula;
`TimesPerWeek` is adjusted until target is met; **[5]**
- (d) *Award up to [2 max].*
Individual and individualistic use:
They don't know each other, and they do their laundry separately, not full load;
They don't cook together and consume more water consequently;
Somebody washes the car every week without telling the others;
- Special occasions:*
Some of them may be athletes and need to wash the sport gear separately;
Somebody invites friends over night/for dinner and the water consumption increases;
- Technical reasons:*
There are water leaks in the house and they are not aware about them;
The counter was manipulated by the landlord years ago; **[2]**

7. (a) *Award up to [4 max] as follows:*
Award [1 max] for identifying an advantage and [1 max] for an expansion.
Award [1 max] for identifying a disadvantage and [1 max] for an expansion.
- Advantages (requires an expansion):*
They may see a complex situation numerous times / gain confidence in using the surgical tools;
Without putting patients at risk / or keep practicing until they feel confident that they will be able to get it right in reality;
- Disadvantages (requires an expansion):*
They don't have real-life experience;
Therefore, we don't know whether they will faint at the sight of blood;
- They may be trained just on the software (a unique body shape);
But body shapes are different and students might find themselves at odds;
- They do not get an idea of how hectic the theatre can be/space available/interferences among people;
And this training alone will be useless; [4]
- (b) *Award up to [3 max].*
It needs sensors to assess the **pressure** of the hand on the instrument;
Its **positioning with respect to the touchscreen** needs to be tracked (for example by an external camera);
Its **inclination** with respect to the touchscreen needs to be tracked;
The **change of tools** held should be quickly reflected in the image that is displayed (affects communication);
The **specific tool** needs to be recognized; [3]
- (c) The representation of (abstract) data;
In a way that is understandable by humans; [2]
- (d) *Award up to [6 max].*
Award [1 max] for identifying an advantage of the use of 3D and [1 max] for an expansion up to [3 max] for discussing the advantages of the use of 3D;
Award [1 max] for identifying a disadvantage of the use of 3D and [1 max] for an expansion up to [3 max] for discussing the advantages of the use of 3D;
Award [1 max] for a considered conclusion;
- Note to examiners: To achieve [6] a conclusion must be provided.*
If there is no conclusion the maximum mark is 5.
- A 3D simulation will be more realistic;
As depth can be shown;
Patient's body can be rotated; // allow up to [2] for example;
- But, a 3D simulation takes more time to render / more calculations have to be made;
So, may not be displayed in real time;
It is more demanding on memory / may exceed available memory;
- For an application, as serious as medical training, adequate hardware and software should be made available so that 3D simulations can be carried out; [6]

8. (a) The initial population set is chosen randomly/pseudo randomly;
A fitness function is applied to each population;
The fittest members are selected for the next stage;
Genetic operators are applied;
Such as crossover / mutation;
The process is repeated until;
An acceptable level of fitness is found / a plateau is reached / a maximum number of iterations has been reached; [6]
- (b) (i) Network networks initially have a randomly chosen set of weights;
Which will not solve any problem;
These weights must be altered;
Until they fulfil their objective; [4]
- (ii) *Award up to [6 max].*
Award [3 max] for considerations linked to each of the two forms of learning.
With supervised learning the processing of the image is guided (supervised) by the output dataset to the purpose of classifying certain features of the image;
For example, to the value of the pixel (grey or colour) one can assign likelihood of belonging to neighbouring values of greys/colours;
And this can be defined in advance as a way to represent a potential value for an infected tree;

However, with unsupervised learning the processing of the image is not guided in any way / the output is not known;
And the purpose is to discover and build clusters that have common characteristics;
Therefore, this may clearly help in building up the main regions, but not specifically on focussing on the infected trees; [6]
- (c) *Award up to [4 max].*
Questions/requests must be short so repetition is not required / so they are immediately understood;
And chosen so that there is a restricted response set;
That can be recognised by the system / that can be stored by the system;
The requests must lead to a satisfactory end;
Or customers will get frustrated / hang up / the company will lose business etc; [4]

Option C — Web science

9. (a) *Award up to [2 max].*
Provide a reference to the source/provider;
Obtain permission from the interviewee; [2]
- (b) *Award up to [2 max].*
To better control access to their servers;
Because it could be cause of bottleneck on the server / to reduce/minimize risks of DDoS;

To better manage the UI;
So that the offer of podcasts is well visible also from small devices;

To have a smaller system that is easier to maintain;
That can be maintained more effectively also remotely by one person; [2]
- (c) *Award up to [1 max].*
(accept for general sound format:) MP3, FLAC, WAV, WMA; [1]
- (d) *Award up to [5 max]:*
Award up to [4 max] for comparing lossless and lossy.
Award [1] for a conclusion with reference to the scenario.

In terms of audio quality for spoken language there is not much difference in using lossy or lossless compression/ lossless compression does not really bring benefit to the spoken audio;

A minor benefit could be in using lossy compression in relation to transmission while downloading (and also in streaming);

And it is relevant, in general, in a situation with low bandwidth;

However, these files are small in size anyway, because their duration is limited to 10 minutes;

And this suggests that for the specific nature and size of the files (spoken and 10 minutes duration) lossy might be slightly preferable for some users for downloading without compromising on quality of audio. [5]

- (e) *Award up to [2 max].
Award [1] for identifying a reason why buffering may occur and [1] expansion/example up to [2 max].*

Too many people are using their own Internet access point at home simultaneously;
Possibly on different devices, including tablets, smartphones on WiFi;

There is low bandwidth because it is peak time;
Possibly too many people are downloading simultaneously;

Old router/modem/computer / outdated software may slow performance;
Computer that is not protected by antivirus may have performance affected by malware;
Performance;

There are too many applications running simultaneously on the computer;
So, memory is insufficient and one should close some applications;

[2]

- (f) *Award up to [4 max].
Award up to [2 max] for the use of dynamic scripting.
Award up to [2 max] for the means of tracking a person's preferences.*

Dynamic web pages / scripting

Podcasts are associated to XML objects;
So, that the list of podcasts can be processed by scripts that can re-arrange the order of the XML objects;
To present them in a way that takes into account the previous uses from the user;
By monitoring their preferences, over time;

Tracking a person's preferences

For example, counting the clicks and themes to extract regularities / support data analytics;
That can be identified for example by the IP of their device when this is not anonymised;
And also stored in the history of the browser/by use of cookies;

[4]

10. (a) *Award up to [2 max].*
 HTTPS authenticates the identity of the website;
 And encrypts data that are transmitted between web browser and web server/between two parties; [2]
- (b) *Award up to [4 max].*
Award [1] for step that the Domain Name System (DNS) server will take in order to locate the correct IP address for this request from the browser up to [4 max].
- The DNS server configured to the browser’s computer / the ISP (Internet Service Provider) checks through its own database to see if the (domain) name is there;
 If it is, it will return the corresponding IP address to the browser;
 If it isn’t the request is passed onto the next DNS server (in the hierarchy);
 This continues until the (domain) name is found;
 Or the top level / authoritative DNS server if reached;
 When IP address is found, it is sent back to the original DNS server; [4]
- (c) *Award up to [2 max].*
 The required layout style for the footer is saves as a CSS file;
 Each time the footer section/element appears it calls this CCS file; [2]
- (d) *Award up to [4 max].*
Award [1] for each comment that indicates the function of the code up to [4 max].
Award [1] realising the modelling centred on a session (intuition of what \$_SESSION[] is)
Award [1] realising that session user identifier shall be unique.
Award [1] realising the existence of users_db.
Award [1] realising that this database is interrogated with this session identifier to retrieve all fields of the record;
Award [1] to instantiate a variable \$userRecord.
- The shopping basket is modelled centred on the “session”;
 A global array \$_SESSION exists, holding a session identifier associated with any user;
 And a new sessionUserID is added to the array and instantiated by a unique value for identifier;
 That is generated by the function new random_number();
 This element of the array \$_SESSION[] is used to retrieve in the user db (users database) the entire record for the user;
 Relative to the user and specific to that particular session;
 And used to instantiate the variable \$userRecord for further processing;
 For example the final billing or confirmation of payment/dispatch. [4]
- (e) *Award up to [2 max].*
 The identifier “id” that is passed in the variable \$_REQUEST['id'];
And the specific action that uses the value “removeBasketItem”; [2]

11. (a) *Award up to [2 max].*
Award [1] for describing the challenge resulting from some limitation and [1] expansion, possibly with example up to [2 max].
Old computers impose memory and processing constraints;
It is likely that they cannot make sophisticated web pages;
For example, they may need to reduce the number of interactions/transmissions;
By deciding not to have dynamic web pages;
To the purpose of not generating loss of processing power on their side;
- The software might not be up-to-date/the OS is old;
This poses a threat for viruses or deliberate attacks (attack the weakest nodes);
To the point that not even the charities will link to their website, rather just listing the name of the association;
To limit the risk of propagating malware; [2]
- (b) *Award up to [4 max].*
Award [1] for each reason how the use of cloud-based services may help Harvest to overcome the limitations of the available technologies up to [4 max].
They can host the website on the cloud;
Enabling a sophisticated site to be created / bypassing their own limitations;
The cloud could store all of their data;
As they will have limited storage on their own equipment; [4]
- (c) *Award up to [4 max].*
Award [1] for identifying how the use of cloud-based services may cause problems and [1] for an expansion / development up to [2 max].
Mark as [2] and [2].
The costs are likely to be high;
And *Harvest* is clearly a company with limited funds;
- The cloud company will be bound by rules and regulations in the country in which it is registered;
These may cause conflicts with regard to the type of information that *Harvest* deals with; [4]

- (d) *Award up to [5 max].
Award [1] for identifying an advantage of using black hat techniques and an additional [1] for each development of that advantage up to [2 max].
Award [1] for identifying a disadvantage of using black hat techniques and an additional [1] for each development of that disadvantage up to [2 max].*

Award [1] for a valid conclusion.

Advantages:

The use of black hat techniques + an example (*eg* hidden links, or reciprocal links, keyword stuffing, hidden content, content theft, *etc*);
May temporarily improve its ranking;

Disadvantages:

However, this is unethical;
This may not sit well with its image as a charity;

Conclusion:

Search engines will eventually discover that these techniques are being used;
This may lead to the site being banned;

[5]

12. (a) *Award up to [4 max].
Award [2 max] for a general knowledge and understanding of multimedia web.
Award [2 max] for describing a situation where the multimedia web may be beneficial to using a text based search.*

Knowledge of multimedia web

The multimedia web makes an expanded use of the semantic web (formally and semantically interlinked data of any kind) making possible to provide input in a form other than text, and possibly return output in various forms;

Therefore, the search is made at the level of “concepts” and not just textual terms;

In a way that the computer can retrieve semantical links among different concepts/concepts presented in different formats;

That includes the ability for crawlers to go through scientific classifications of text/captions/description of images;

That can all be indexed and clustered into a common “concept”;

Example 1:

To detect/discover/verify/monitor the presence of species based on sound

For example:

We might aim in recognising and searching on the web starting from the sound of some nocturnal animal that we don't see/we don't know, but whose presence is revealed by their eyes;

Giving the sound as input would allow to perform a search based on registration/audio that would run on multimedia sources linked with semantical links;

Example 2:

To detect/discover/verify/monitor the presence of species based on images

For example:

We might start a multimedia search based on images/photos of nocturnal animals we may have taken;

To see whether those sightings were associated to dangerous species or not;

To signal the presence of some unusual species in the region;

To signal the presence of some animal in difficulty to the protection agency;

[4]

- (b) *Award up to [3 max].
Award [1] for each difference up to [3 max].*

Designed or extracted by knowledge engineers VS;

Created collaboratively by users;

Laborious and expert VS quick/easy non-expert;

Controlled vocabulary/dictionary VS no control of vocabulary/informal;

Formal specification of knowledge domain (taxonomies) VS;

Informal metadata on documents;

Not just for the web VS mostly for the web;

Essential for the semantic web VS important for social web;

Explicit meaning and high expressive power VS;

Ambiguity and low expressive power;

Experts' view of the domain VS social aspect of meaning;

[3]

- (c) *Award up to [6 max].*
Award [2 max] for a superficial response that uses generic terminology.
Award [4 max] for a response that shows some analytical comments, unsubstantiated conclusions and some use of appropriate terminology.
Award [6 max] for a coherent analysis leading to substantiated conclusions with the appropriate use of subject specific terminology.

General, possibly described through examples

The ontology of the semantics web is extracted with expert means from documents;

(For example, by making features analysis / aggregating concepts / eliminating disambiguation;)

Whereas, the folksonomy is the result of adding different vocabulary (done by the users) via tags in posts/web pages/social web (like *Tumblr*);

(For example, tagging the photo of a cat under an informal “cat’s eye” would add a different perspective to the ontology);

The problem:

If not monitored, tagging may be source of “garbage” and hinder the research for the ontology;

The ambiguity problem of tagging needs to be addressed so to extract a weight to ambiguous use of tags;

Possible countermeasures:

Expand the query within social tagging platforms;

In order to extract a weight to ambiguous use of tags and use this weight to cut-out some irrelevant retrieved information;

OR

Use the folksonomy tags (at least some) to update (possibly as metadata) the ontology vocabulary;

However, depending on the context, this may not always be feasible/applicable and depends on the domain of discourse;

OR

Re-engineer the ontology to include a collaborative form of construction from tagging;

However, this is method is not applicable in expert fields where quality of content must be preserved;

OR

Keep the ontology and folksonomy well separated, but let them work in tandem; In a way to have feedback from folksonomies, but merged in a controlled way;

[6]

- (d) *Award up to [4 max].*
Award [1] for each comment that indicates the role that web crawlers perform in developing an index that is used by a search engine up to [4 max].

A web crawler is given a “seed” page which is where it starts;

It searches for hyperlinks;

Which it recursively follows;

Making a copy of each web page visited;

Which can be indexed by the search engine;

[4]

- (e) *Award up to [3 max].*
Award [1] some example to fix the scenario of open source;
Award [1] what is meant by collective intelligence;
Award [1] some explanation/expansion to link the concepts;

The focus:

“Software for Public Interest”;

Open source and free software development is the result of initiatives taken by sparse communities of programmers that collaborate effectively to maintain complex programs/packages of code;

(For example, Debian Foundation, Linux Foundation, Mozilla Foundation, Apache Foundation, Free Software Foundation, Wikipedia...;)

The structure:

A distributed system of development AND tightly governed /

Each developer codes individually, but the whole project relies on the fact that portions of code may be integrated as components of other parts;

Organisations may have different modes of governance, and have precise ways (processes) to contribute to a project, both technical and social;

The objective is to guarantee coordination, quality and relevance for the project development (technical) based on contributions of the individuals;
While modulating possible clashes/conflicts between different personalities;

Examples:

OS kernels, Wikipedia, mail programs, GNU, ...;

Build a deposit of old software (free/open source), for preservation and sharing (with everybody);

Old open source SW may in fact be at risk of not being publicly available for a variety of reasons, such as sites become proprietary and limit public access, crash of the systems, business decisions;

Role of collective intelligence

Individuals are expert developers, but act collectively, for public interest, and this becomes a movement that may make a cultural/political difference/impact (the key is high quality, not populism!);

eg the French Public Administration adopts Open Source software and has rejected proprietary products (Microsoft, specifically)

[3]

Option D — Object-oriented programming

13. (a) (i) *Award up to [1] for each part of a suitable definition up to [2 max].*
An object is an abstract entity;
and its components are data and/or actions; [2]
- (ii) *Award up to [1] for identifying each disadvantage and [1] for an elaboration of this disadvantage, up to [2 max].*
Mark as [2] and [2]. [4]
Unsuitable for minor projects;
since OOP increases complexity for little gain;

OOP programs are larger than other programs;
and therefore slower;

OOP programs take more effort to construct;
because of the decomposition needed to achieve abstraction;
- (iii) *Award [1] for identifying an advantage and [1] for an elaboration of this advantage up to [2 max].* [2]
Example answers:
Faster development;
Because different programming teams can work on different modules;

Easier to debug;
Because the smaller modules will have fewer mistakes than one big program;

Easier to update (in the future);
Because it is easier to update a module than the full program;

Re-usability;
Modules can be stored in libraries and reused for different programs;
- (b) *Award up to [1] for any indication of aggregation.*
Example answer:
The `Arrival` object has/stores a `Flight` object; [1]

- (c) **Award up to [4 max].**
Award [1] for including 3 distinct sections.
Award [1] for including a component with all variables.
Award [1] for including a component with all listed methods.
Award [1] for indicating private/public using + / -.
Don't penalize the absence of accessor and mutator methods.

[4]

Example answer:

```

class Arrival
- Flight: myFlight
- String: STA
- int: runway
- String: gate
- int: delay
- boolean: landed

+ void addDelay (int)
+ String getETA ()
+ int compareTo (String)
+ int compareTo (Arrival)

```

- (d) **Award up to [4 max].**
Award [1] for a while loop.
Award [1] for correctly testing the flightID using compareTo().
Award [1] for incrementing the loop counter.
Award [1] for a return statement.

[4]

Example answer:

```

private int search(String flightID)
{
    int i = 0;
    while (inbound[i].compareTo(flightID) != 0)
    {
        i++;
    }
    return i;
}

```

- (e) (i) **Award up to [2 max].**
Award [1] for identifying an advantage of using a binary search and [1] for an elaboration of the advantage up to [2 max].

Binary search is much faster than sequential search;
 Because it halves the search range for every comparison;

[2]

- (ii) **Award up to [2 max].**
Award [1] for identifying a disadvantage of using a binary search and [1] for an elaboration of the disadvantage up to [2 max].

However, it is not applicable to unsorted data sets;
 Because the data must be sorted first which adds to computational cost;

[2]

14. (a) *Award up to [2 max].*
Award [1] for stating encapsulation as an OOP property.
Award [1] for an elaboration. [2]

Example answer:

Encapsulation allows to make instance variables (and methods) of a class private to that class;
 So that the main program / other classes can't accidentally access / change the data in an object;

- (b) (i) *Award up to [2 max].*
 The two methods have different parameters;
 allowing the compiler to choose the correct one. [2]
- (ii) Polymorphism [1]

- (c) *Award up to [3 max].*
Award [1] for correctly using `compareTo`.
Award [1] for correctly using `getETA()` twice (with or without "this").
Award [1] for returning the correct result. [3]

Example answers:

```
public int compareTo(Arrival anotherArrival)
{
    int result;
    result = this.getETA().compareTo(anotherArrival.getETA());
    return result;
}
```

```
public int compareTo(Arrival anotherArrival)
{
    String currentETA = this.getETA();
    String otherETA = anotherArrival.getETA();
    return currentETA.compareTo(otherETA);
}
```

15. (a) Award up to **[2 max]**.
[1 max] for any two out of three underlined key components.
[2 max] for all three key components.
Note: the return type is not part of the method signature.

Example answers:

The method name and all of its parameters and the type of these parameters.
 The method name and the type of all of its parameters.

[2]

- (b) Award up to **[4 max]**.
 Award **[1]** for a correct loop until last Arrival.
 Award **[1]** for including correct test for ETA.
 Award **[1]** for including correct test landed.
 Award **[1]** for correctly using double dot notation in output statement.
 Do not penalize incorrect use of accessor methods.

[4]

Example answers:

```
public void showDelayed(String t)
{
    int i = 0;
    while (i <= last)
    {
        if (inbound[i].getETA().compareTo(t) < 0)
        {
            if (!inbound[i].getLanded())
            {
                output(inbound[i].getMyFlight().getID());
            }
        }
        i = i + 1
    }
}

public void showDelayed(String t)
{
    int i = 0;
    while ((i <= last) && (inbound[i].getETA().compareTo(t) < 0))
    {
        if (!inbound[i].landed)
        {
            output(inbound[i].myFlight.id);
        }
        i = i + 1
    }
}

public void showDelayed(String t)
{
    for (int i = 0; i <= last; i++)
    {
        if ((inbound[i].getETA().compareTo(t) < 0) &&
            (!inbound[i].landed))
        {
            output(inbound[i].myFlight.id);
        }
    }
}
```

- (c) **Award up to [6 max].**
Award [1] for a loop comparing *newArrival* to entries in *inbound*.
Award [1] for testing for the last entry.
Award [1] for good attempt a for loop in reverse order.
Award [1] for a fully correct for loop in reverse order.
Award [1] for correctly shifting elements in the array.
Award [1] for assigning *newArrival*.
Award [1] for incrementing *last*.

[6]

Example answer:

```
public void add(Arrival newArrival)
{
    int i = 0;
    while ((i <= last) && (inbound[i].compareTo(newArrival) < 0))
    { i++;
    }
    for (int j = last + 1; j > i; j--)
    { inbound[j] = inbound[j - 1];
    }
    inbound[i] = newArrival;
    last++;
}
```

Alternative answer:

```
public void add(Arrival newArrival)
{
    int i = 0;
    while (i <= last) &&
        (inbound[i].getETA().compareTo(newArrival.getETA()) < 0)
    { i++;
    }
    for (int j = last; j >= i; j--)
    { inbound[j + 1] = inbound[j];
    }
    inbound[i] = newArrival;
    last++;
}
```

- (d) **Award up to [4 max].**
Award [1] for declaring a variable of type *Arrival*.
Award [1] for calling *remove(flightID)*.
Award [1] for calling *addDelay(minutes)*.
Award [1] for adding the updated *Arrival* object to array *inbound*.
Award [1] for adding the update to the array *inbound* by calling *add(update)*.

[4]

```
public void delay(String flightID, int minutes)
{
    Arrival update = remove(flightID);
    update.addDelay(minutes);
    add(update);
}
```

16. (a) Award up to [2 max].
 Award [1] for identifying an advantage.
 Award [1] for an elaboration of the advantage. [2]

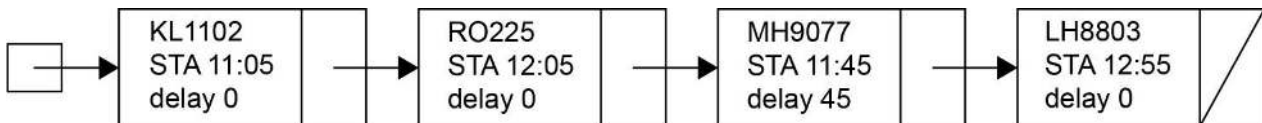
Example answers:

Convenience;
 Because implementations of common tasks are available;
 Reliability;
 because these implementations are fully developed, functional and robust.

- (b) Award up to [4 max].
 Award [1] for identifying an issue and [1] for a valid comparison related to the issue.
 Mark as [2] and [2] [4]

Issue	dynamic linked list	static array
memory use	efficiently uses memory as needed	uses a fixed amount of memory which is inefficient
access	only sequential access	direct access using an array index
overflow	amount of elements only limited by amount of memory available	overflow is a problem as there is a fixed amount of elements allowed
manipulation	rearranging is easy by changing pointers	rearranging often involves shifting elements

- (c) Award up to [2 max].
 Award [1] for a good attempt (for example the new node being added as the 3rd node in the list).
 Award [2] for a fully correct diagram. [2]



(d) (i) When a method calls on itself. [1]

(ii) **Award up to [4 max].**
*Award [1] for correct recursion levels i including $i = 3$.
Award [1] for the correct flight id's (with or without the line for $i = 3$.
Award [1] for 12:55 and 11:05 in the correct order.
Award [1] for 12:30 (ETA) in the correct order.*
Note: that there are many ways to trace a recursive algorithm. [4]

i	id	output
0	KL1102	
1	MH9077	
2	LH8803	
3	-	
2	LH8803	12:55
1	MH9077	12:30
0	KL1102	11:05

(e) For answers without iterators

Award up to [7 max].

Award [1] for correctly instantiating the `LinkedList` variable `result`.

Award [1] for declaring (and instantiating) all other variables.

Award [1] for correct main loop.

Award [1] for correctly comparing the first Arrivals in the two lists.

Award [1] for correctly removing the earlier Arrival from its original list.

Award [1] for correctly adding this Arrival to the end of the resulting list.

Award [1] for attempting to add the remainder of a non-empty linked list.

Award [1] for correct loop for copying the remainder of `runway1`.

Award [1] for correctly removing/adding the remainder of `runway1`.

Award [1] for also copying a possible remainder of `runway2`.

Example without iterator:

```
public LinkedList<Arrival> mergeLists()
{
    LinkedList<Arrival> result = new LinkedList<Arrival>();
    Arrival temp = null;
    while (!runway1.isEmpty() && !runway2.isEmpty())
    {
        if (runway1.getFirst().compareTo(runway2.getFirst()) < 0)
        { temp = runway1.removeFirst(); }
        else
        { temp = runway2.removeFirst(); }
        result.addLast(temp);
    }

    while (!runway1.isEmpty())
    { temp = runway1.removeFirst();
      result.addLast(temp);
    }
    while (!runway2.isEmpty())
    { temp = runway2.removeFirst();
      result.addLast(temp);
    }

    return result;
}
```

Alternative example without iterator:

```
public LinkedList<Arrival> mergeLists()
{
    LinkedList<Arrival> result = new LinkedList<Arrival>();
    Arrival head1 = runway1.peekFirst();
    Arrival head2 = runway2.peekFirst();

    while ((head1 != null) && (head2 != null))
    { if (head1.compareTo(head2) <= 0) // or <
      { result.addLast(head1);
        runway1.removeFirst();
        head1 = runway1.peekFirst();
      }
      else
      { result.addLast(head2);
        runway2.removeFirst();
        head2 = runway2.peekFirst();
      }
    }
    while (!runway1.isEmpty())
```



```
        {
            result.addLast(runway1.removeFirst());
        }
    while (!runway2.isEmpty())
        {
            result.addLast(runway2.removeFirst());
        }
    return result;
}
```

[7]

For answers that use iterators

Award up to [7 max].

Award [1] for correctly instantiating the `LinkedList` variable `result`.

Award [1] for declaring and instantiating iterators and all other variables.

Award [1] for correct main loop.

Award [1] for correctly comparing the first Arrivals in the two lists.

Award [1] for correctly adding the earlier Arrival to the end of the resulting list.

Award [1] for testing `iter.hasNext()` and loop termination.

Award [1] for attempting to add the remainder of a non-empty linked list.

Award [1] for correct loop used for copying the remainder of `runway1`.

Award [1] for correctly copying the remainder of `runway1`.

Award [1] for also copying a possible remainder of `runway2`.

Example with iterator:

```
public LinkedList<Arrival> mergeLists()
{
    LinkedList<Arrival> result = new LinkedList<Arrival>();
    ListIterator<Arrival> iter1 = runway1.listIterator();
    ListIterator<Arrival> iter2 = runway2.listIterator();
    Arrival curr1 = null;
    Arrival curr2 = null;
    if (iter1.hasNext()) {curr1 = iter1.next();}
    if (iter2.hasNext()) {curr2 = iter2.next();}

    while ((curr1 != null) && (curr2 != null))
    {
        if (curr1.compareTo(curr2) <= 0)           // or <
        {
            result.addLast(curr1);
            if (iter1.hasNext()) {curr1 = iter1.next();}
            else {curr1 = null;}
        }
        else
        {
            result.addLast(curr2);
            if (iter2.hasNext()) {curr2 = iter2.next();}
            else {curr2 = null;}
        }
    }

    while (iter1.hasNext()) result.addLast(iter1.next());
    while (iter2.hasNext()) result.addLast(iter2.next());
    return result;
}
```
